



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

www.bdigital.ula.ve
MÓDULO EDUCACIONAL PARA IMPLEMENTACIÓN DE
PLANTAS ANALÓGICAS Y CONTROLADOR PID

Br. Ulises Goudier Ramírez Kuoman

Mérida, Mayo, 2022

Reconocimiento-No comercial-Compartir igual



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

MÓDULO EDUCACIONAL PARA IMPLEMENTACIÓN DE
PLANTAS ANALÓGICAS Y CONTROLADOR PID

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero
Electricista

Br. Ulises Goudier Ramírez Kuoman
Tutor: MSc. Francisco J. Viloría M.

Mérida, Mayo, 2022

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

**MÓDULO EDUCACIONAL PARA IMPLEMENTACIÓN DE PLANTAS
ANALÓGICAS Y CONTROLADOR PID**

Br. Ulises Goudier Ramírez Kuoman

Trabajo de Grado, presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

Prof. Oscar E. Blanco O.

Prof. Orlando C. Ostos C.

Prof. Francisco J. Viloría M.

DEDICATORIA

El presente trabajo de grado está dedicado a mis padres, hermana, sobrina, hijo y tutor, los cuales me han apoyado incondicionalmente para lograr cumplir esta meta, de todo corazón les dedico este logro.

www.bdigital.ula.ve

AGRADECIMIENTOS

A mis docentes y en especial a mi tutor el profesor Fráncico Viloría por su ayuda, paciencia y dedicación en este trabajo de grado de verdad eternamente agradecido, a mi madre Marisol Kuoman por siempre apoyarme con sus palabras de aliento, a mi padre José Luis Ramírez por su apoyo incondicional que me ha brindado durante toda la carrera sin desistir, a mi hermana Natalia por estar siempre presente al igual que mi sobrina Julieta, a mi hijo Roosevelt por hacer de mí una mejor persona y acompañarme en mis sueños y metas.

Asimismo, quiero agradecer a mis amistades: Álvaro Ramírez, Gabriel Montilla, Carlos Moreno, Antonio Valbuena, Lisbeth Espinoza, Sanua Radouan, Juan Manuel Figueroa y Aarón Contreras por su amistad y apoyo incondicional durante toda mi formación como ingeniero electricista.

Ulises Goudier Ramírez Kuoman. Módulo educacional para implementación de plantas analógicas y controlador PID. Universidad de Los Andes. Tutor: MSc. Francisco J. Vilorio M. Mayo, 2022.

RESUMEN

El presente proyecto describe la implementación de un módulo educacional para sistemas de control, este emulador digital de plantas industriales con un controlador PID digital está orientado al ámbito académico. El sistema se usa por medio de una interface gráfica de usuario (página web) intuitiva y fácil de usar, con el propósito de ayudar y orientar a los estudiantes, reemplazando el uso de plantas analógicas a escala, permitiendo simular el equivalente de sistemas industriales de forma digital, de tal manera que los estudiantes sean capaces de practicar y entrenarse en los métodos de control, para ello se hizo el estudio de la teoría de controladores PID, discretización de sistemas para convertir estos procesos de forma digital y se implementó el software necesario y más adecuado para simular estos procesos en un microcontrolador ESP32. Se realizaron las simulaciones y pruebas pertinentes analizando los resultados obtenidos.

Descriptor: Microcontrolador ESP32, Controladores PID, Emulador Digital.

INDICE GENERAL

DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
RESUMEN.....	vi
INDICE GENERAL.....	vii
INDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS.....	xiii
INTRODUCCIÓN.....	1
CAPITULO 1 PROPUESTA DE TRABAJO.....	2
1.1 JUSTIFICACIÓN.....	2
1.2 OBJETIVOS.....	3
1.2.1 Objetivo general.....	3
1.2.2 Objetivos específicos.....	3
1.3 METODOLOGÍA.....	3
1.4 ALCANCE.....	3
1.5 LIMITACIONES.....	4
CAPITULO 2 MARCO TEÓRICO.....	5
2.1 PLANTAS.....	5
2.2 TIEMPO MUERTO.....	6
2.2.1 Aproximaciones polinomiales de tiempo muerto.....	7
2.2.2 Aproximación por Padé.....	7
2.3 PERTURBACIONES.....	9
2.4 CONTROLADORES PID.....	10
2.5 SINTONIZACION DE CONTROLADORES PID.....	11

2.5.1	Método de Ziegler-Nichols a lazo abierto	12
2.5.2	Método de Dahlin a lazo abierto	14
2.6	TRANSFORMADA Z.....	14
2.6.1	Métodos de la transformada Z.....	16
2.7	CONTROLADOR PID DISCRETIZADO.....	18
CAPITULO 3 DISCRETIZACION DE LOS SISTEMAS.....		20
3.1	DISCRETIZACION DE LAS PLANTAS	20
3.1.1	Plantas con tiempo muerto	25
3.1.2	Controlador PID	31
3.1.3	Controlador PID en tiempo continuo	31
3.1.4	Controlador PID en discreto.....	34
CAPITULO 4 IMPLEMENTACION DEL CODIGO EN MATLAB Y MICRO		38
4.1	IMPLEMENTACION DEL CODIGO EN MATLAB.....	38
4.2	MICROCONTROLADOR ESP32	44
4.2.1	Características del micro ESP32.....	45
4.2.2	Interfaces periféricas.....	45
4.3	IMPLEMENTACION DEL CÓDIGO PARA EL MICRO ESP32.....	45
4.4	INTERFAZ GRAFICA.....	66
4.4.1	Diseño de página web.....	66
4.5	COMUNICACION ENTRE ESP32 Y PAGINA WEB	67
4.5.1	Notación de objeto de JavaScript (<i>Json</i>)	67
4.5.2	Protocolo de comunicación <i>WebSockets</i>	69
4.6	IMPLENTACION DEL CODIGO JAVASCRIPT	69
CAPITULO 5 ANALISIS DE RESULTADOS		79
5.1	RESULTADO DEL MODULO EDUCACIONAL	79

5.2	EJEMPLO 1	83
5.3	EJEMPLO 2	84
	CONCLUSIONES	87
	RECOMENDACIONES	88
	REFERENCIAS	89
	ANEXOS	91
	MANUAL DEL USUARIO	92

www.bdigital.ula.ve

INDICE DE FIGURAS

Figura 2.1 Respuesta del sistema, salida vs entrada con tiempo muerto.....	6
Figura 2.2 Diagrama de bloques de un sistema de control a lazo cerrado con perturbaciones ...	9
Figura 2.3 Respuesta ante un escalón.....	12
Figura 2.4 Respuesta de un escalón unitario para sintonización del controlador.....	13
Figura 2.5 Gráfica de precisión y complejidad de los métodos de la transformada Z.	17
Figura 3.1 Comparación de las respuestas en tiempo continuo y discreto de la planta integrante.	21
Figura 3.2 Respuesta de la planta integrante en tiempo continuo y discreto.....	21
Figura 3.3 Comparación de las respuestas en tiempo continuo y discreto. Planta de primer orden.	22
Figura 3.4 Respuesta de la planta de primer orden en tiempo continuo y discreto.	22
Figura 3.5 Comparación de respuesta de la planta de primer orden integrante en tiempo continuo y discreto.	23
Figura 3.6 Respuesta de la planta de primer orden integrante en tiempo continuo y discreto.	23
Figura 3.7 Comparación en tiempo continuo y discreto de la planta de segundo orden.	24
Figura 3.8 Respuesta de la planta de segundo orden en tiempo continuo y discreto.	24
Figura 3.9 Comparación de la planta integrante con tiempo muerto de segundo orden.	26
Figura 3.10 Respuesta de la planta integrante con tiempo muerto.	27
Figura 3.11 Respuesta de la planta de primer orden con tiempo muerto.	28
Figura 3.12 Respuesta de la planta de primer orden integrante con tiempo muerto.	28
Figura 3.13 Respuesta de la planta de segundo orden con tiempo muerto.....	29
Figura 3.14 Respuesta por el método de aproximación de Padé.....	30
Figura 3.15 Respuesta de tiempo muerto por desplazamiento de líneas de retardo.....	31
Figura 3.16 Diagrama del controlador PID a lazo cerrado.....	31
Figura 3.17 Respuesta a lazo cerrado del controlador para una planta integrante.	32
Figura 3.18 Respuesta a lazo cerrado del controlador con planta de primer orden.....	32
Figura 3.19 Respuesta a lazo cerrado del controlador PID de planta de primer orden integrante.	33

Figura 3.20 Respuesta a lazo cerrado del controlador PID para planta de segundo orden.	33
Figura 3.21 Diagrama del controlador PID discreto a lazo cerrado.	34
Figura 3.22 Respuesta del controlador PID discreto de la planta integrante.....	35
Figura 3.23 Respuesta del controlador PID discreto de la planta de primer orden.	35
Figura 3.24 Respuesta del controlador PID discreto de la planta de primer orden integrante. .	36
Figura 3.25 Respuesta del controlador PID discreto de la planta de segundo orden.	37
Figura 4.1 Respuesta de la planta integrante en código de Matlab.	40
Figura 4.2 Respuesta de la planta de primer orden en código de Matlab.....	41
Figura 4.3 Gráfica de la planta de primer orden integrante en código de Matlab.....	42
Figura 4.4 Respuesta de la planta integrante con tiempo muerto.....	43
Figura 4.5 Respuesta de la planta de primer orden con tiempo muerto.	43
Figura 4.6 Respuesta de la planta de primer orden integrante con tiempo muerto.	44
Figura 4.7 Inicialización de <i>WebSockets</i>	46
Figura 4.8 Conexión al router.....	47
Figura 4.9 Conexión como punto de absceso al módulo ESP32.....	48
Figura 4.10 Inicialización del servidor.....	49
Figura 4.11 Inicialización del <i>SPIFFS</i>	50
Figura 4.12 Función <i>setup</i>	51
Figura 4.13 Función <i>loop</i>	52
Figura 4.14 Función <i>flags</i> para ejecutar tipo de planta.	53
Figura 4.15 Cálculo de la salida de la planta con o sin perturbaciones.....	54
Figura 4.16 Función de cálculos de las plantas y desplazamiento.	55
Figura 4.17 Cálculo de variables de las plantas y reserva de memoria.....	56
Figura 4.18 Función de desplazamiento por vectores.	57
Figura 4.19 Cálculo del controlador PID.	58
Figura 4.20 Envío de lecturas para la gráfica.	59
Figura 4.21 Petición de parámetros de la interfaz.	60
Figura 4.22 Código del botón para iniciar y detener la simulación.....	61
Figura 4.23 Acción de empezar y detener el programa.....	62
Figura 4.24 Asignación de perturbación.	63
Figura 4.25 Asignación de variables de las plantas.....	64

Figura 4.26 Asignación de los valores del controlador PID.....	65
Figura 4.27 Diagrama de secuencia de intercambio de mensajes.	68
Figura 4.28 Envío del status del gpio.	69
Figura 4.29 Envío de los valores del controlador PID.	70
Figura 4.30 Envío del valor de la referencia a través de <i>sliders</i>	71
Figura 4.31 Envío del id de <i>doaction</i>	72
Figura 4.32 Actualización de los campos y asignación de lo que llega al <i>Json</i>	73
Figura 4.33 Evento de nuevas lecturas y envío por <i>Json</i>	74
Figura 4.34 Evento del sensor de temperatura para la gráfica.....	75
Figura 4.35 Evento <i>open</i> de conexión de la página.....	75
Figura 4.36 Evento de erro al desconectarse la página.....	76
Figura 4.37 Impresión de mensajes por <i>JavaScript</i>	76
Figura 4.38 Evento de nuevas lecturas y grafica de los puntos.....	77
Figura 4.39 Asignación y envío de parámetros por <i>Json</i>	78
Figura 5.1 Comunicación <i>WebSockets</i> establecida.....	79
Figura 5.2 Interfaz grafica con página web.....	80
Figura 5.3 Ventana emergente para ingresar parámetros de la planta integrante.....	81
Figura 5.4 Ventana emergente para ingresar los parámetros de la planta de primer orden.....	81
Figura 5.5 Ventana emergente para ingresar parámetros de la planta de segundo orden.....	82
Figura 5.6 Registro de plantas y parámetros.	82
Figura 5.7 Simulación del controlador PID con la referencia.	83
Figura 5.8 Respuesta de sistema sintonizado por el método de Shinkey.	84
Figura 5.9 Respuesta del sistema sintonizado por el método de Ziegler-Nichols.....	85
Figura 5.10 Respuesta del sistema ajustando los parámetros.	86
Figura 5.11 Respuesta del sistema ajustando los parámetros.	86

ÍNDICE DE TABLAS

Tabla 2.1 Parámetros de sintonización por el método de Ziegler-Nichols a lazo abierto.	13
Tabla 2.2 Parámetros de sintonización por el método de Dahlin a lazo abierto.....	14
Tabla 2.3 Algunas transformadas Z.....	15

www.bdigital.ula.ve

INTRODUCCIÓN

Los controladores PID son los más utilizados en la industria, debido a que su función es mejorar la respuesta de cualquier proceso, este trabajo de grado simula los procesos de un controlador PID con 4 modelados de plantas de manera digital a través de una interfaz gráfica en una página web, utilizando un microcontrolador ESP32 al cual se le implementó un código que simula los procesos de las plantas como también del controlador PID.

Este trabajo está enfocado en el ámbito educativo con el fin de que los estudiantes obtengan una herramienta digital, de bajo costo que simule las plantas y el controlador PID, reemplazando los sistemas a escala, los cuales constan de equipos para trabajar solo en el laboratorio, aunque existen sistemas basados en *hardware in the loop* (HIL) con sensores y actuadores que simulan el funcionamiento de las plantas, la diferencia, es que se trabaja de forma digital, el software simula los procesos de 4 tipo de plantas y del controlador PID a través de un microcontrolador ESP32 por medio de un menú o interface de página web, la cual se conecta a un servidor.

Los capítulos que se presentan más adelante exponen el proceso que se realizó para llevar a cabo este proyecto, el capítulo 2 plantea la base teórica necesaria de controladores PID, sintonización y discretización de sistemas. En el capítulo 3 se refleja la discretización de las plantas y controlador PID realizando las pruebas para analizar los resultados a través de Simulink Matlab y por último el capítulo 4, donde se hizo la implementación del código que simula los procesos por Arduino y se toma de referencia el código implementado en Matlab.

CAPITULO 1

PROPUESTA DE TRABAJO

En este capítulo, se plantea la implementación de un módulo educacional de controladores PID digital, con la finalidad de ayudar a los estudiantes de la escuela de ingeniería eléctrica de la Universidad de los Andes en el área de control, a comprender el funcionamiento del controlador PID, a través de esta herramienta de bajo costo, podrá simular cuatro procesos discretos a través de una interfaz intuitiva y de fácil uso desde cualquier lugar, solo necesitará una computadora, un servidor (*router*) y el microprocesador ESP32 el cual tendrá el software que emula estos procesos.

1.1 JUSTIFICACIÓN

La mayoría de los procesos industriales cuentan con una serie de lazos de control con variables como la temperatura, presión, flujo y velocidad entre otros, que necesitan ser ajustados dentro de un rango de operación. La sintonización de los controladores PID, consiste en la determinación del ajuste de sus parámetros (K_p , T_i , T_d), para lograr un comportamiento del sistema de control aceptable y robusto de conformidad con algún criterio de desempeño establecido. Para realizar la sintonización de los controladores, primero debe identificarse la dinámica del proceso, y a partir de ésta, determinar los parámetros del controlador utilizando el método de sintonización seleccionado. Por otra parte, para el estudio de los sistemas de control, se hace necesario aplicar en sistemas reales los conocimientos adquiridos para lograr fortalecerlos, es por esto que se realiza la implementación de un módulo que emulen el comportamiento de las plantas o sus funciones de transferencias, y emular el comportamiento del controlador PID.

1.2 OBJETIVOS

1.2.1 Objetivo general

Construir un prototipo que emule el comportamiento de un sistema de control compuesto por un controlador y una planta.

1.2.2 Objetivos específicos

- Implementar el controlador PID en su versión digital.
- Simular el funcionamiento de planta digital, al menos en 4 versiones que más se encuentran en la industria.
- Implementar el sistema en un sistema de desarrollo como el ESP32 que tiene suficientes recursos y puede usarse con una interfaz amigable.
- Desarrollar una interfaz intuitiva y fácil de usar.
- Elaborar el manual de operación y el informe.

1.3 METODOLOGÍA

La metodología se basa en una revisión bibliográfica del tema, donde el estudiante investigue sobre dicha tecnología. Adicionalmente a través de simulaciones se logrará comprender completamente el modelo funcional. Finalmente, mediante la implementación de la planta se logrará verificar el funcionamiento del sistema implementado usando un controlador PID digital.

1.4 ALCANCE

Se sabe que existe un sin número de modelados industriales (plantas), para este trabajo solo se usaran 4 tipos de plantas con un controlador PID digital, todo en un mismo sistema con una interfaz gráfica (página web) y una sola perturbación.

1.5 LIMITACIONES

La implementación de este proyecto no permite la actuación de simular varias plantas al mismo tiempo, solo permite la ejecución de una planta a la vez. El servidor solo permite que trabaje un usuario a la vez, aunque permita el acceso a múltiples usuarios, solo uno de ellos controlará la simulación. Aunque existe una perturbación en el sistema, no se puede emular perturbaciones complejas en ella.

www.bdigital.ula.ve

CAPITULO 2

MARCO TEÓRICO

En este capítulo se describe la teoría de los controladores PID, tipos de sintonización y los objetivos de las operaciones proporcional, integral y derivativa. Los métodos de discretización usados y modelamientos de las plantas utilizadas.

2.1 PLANTAS

Una planta es el modelado de un dispositivo o el grupo de elementos que posee, los cuales funcionan en conjunto y realizan una operación en particular (como un horno microondas, dispositivo mecánico, un reactor o un avión). Para este trabajo se utilizarán cuatro tipos de plantas: la integrante (ecuación (2.1) la cual hace referencia a la dinámica del llenado de un tanque; de primer orden (ecuación (2.2) representa el modelado de un reactor, pero si se quiere observar la salida del reactor por medio de una tubería, se adquiere el tiempo muerto del sistema; de primer orden integrante (ecuación (2.3) es aplicado en control de velocidad de motores de inducción y por última, la planta de segundo orden (ecuación (2.4), esta planta también puede verse en un reactor, pero depende de la configuración del mismo, al incluir más variables en el sistema.

La razón por la cual se escogieron estas plantas es porque su modelado son los más usados en la industria.

$$G(s) = \frac{K}{s} \quad (2.1)$$

$$G(s) = \frac{K\tau}{\tau s + 1} \quad (2.2)$$

$$G(s) = \frac{K\tau}{(\tau s + 1)s} \quad (2.3)$$

$$G(s) = \frac{K\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.4)$$

2.2 TIEMPO MUERTO

En el área de control a lazo cerrado, el tiempo muerto t_o , es el tiempo que existe entre el cambio de la entrada de un sistema y la respuesta de salida del mismo. Este tiempo se produce en un cambio ocurrido en alguna variable de entrada del sistema y se puede percibir en el dispositivo de medición.

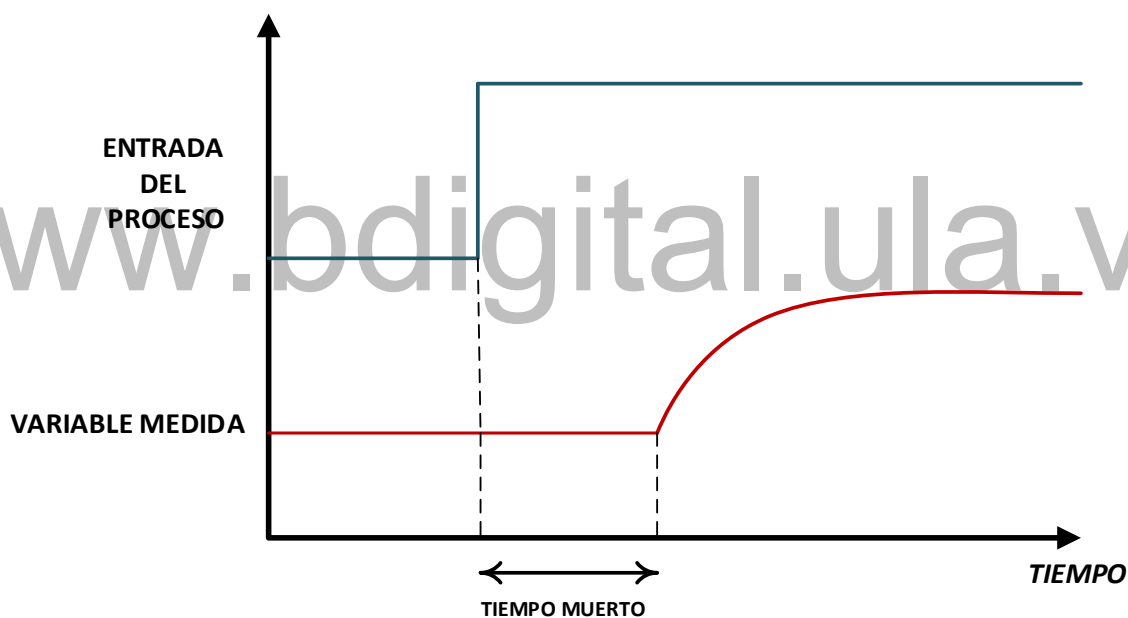


Figura 2.1 Respuesta del sistema, salida vs entrada con tiempo muerto.

El tiempo muerto viene representado en sistemas lineales continuos invariantes en el tiempo como $e^{-\theta s}$.

2.2.1 Aproximaciones polinomiales de tiempo muerto.

El tiempo muerto puede representarse directamente en el dominio de la frecuencia, esto se debe a que los métodos de análisis y diseños, pueden usarse sin aproximaciones en procesos de tiempo muerto. Aunque la función de transferencia del tiempo muerto no es racional, si se requieren representarlas en forma de polos y ceros, por ejemplo, cuando se utilizan los métodos como lugar de raíces o ubicación de polos, si es necesario usar aproximaciones polinómicas de tiempo muerto [1].

Existen varias aproximaciones polinomiales de tiempo muerto, como la expansión por series de Taylor, función de transferencia de retardo múltiple LAG y aproximación de Padé.

La aproximación de Padé es la más utilizada para aproximar el termino no racional del retardo a expresiones polinómicas [1].

2.2.2 Aproximación por Padé

Es la aproximación de una función racional de un orden dado. La serie de potencias coincide con la serie de potencias de la función que se aproxima. Esta aproximación, da un mejor acercamiento de la función que truncan la serie de Taylor, y funciona incluso donde esta no es convergente. Por esta razón se usa ampliamente en los cálculos de computadoras [1].

Si se tiene una función cualquiera $F(x)$, dicha aproximación debe coincidir con el desarrollo de la serie de Taylor de la misma función, la cual se podría llamar $R(x)$, como se muestra en la ecuación (2.5).

$$F(x) \approx R(x) \quad (2.5)$$

En la ecuación (2.5) la idea es hacer que las funciones sean iguales en puntos específicos, es decir, se analiza la función cuando se sustituye $x = 0$ y también cuando se derive la función y x se iguale a cero, debe corresponder al mismo punto logrando las ecuaciones (2.6), (2.7) y (2.8).

$$f(0) = R(0), \quad (2.6)$$

$$f'(0) = R'(0), \quad (2.7)$$

$$f''(0) = R''(0) \quad (2.8)$$

Para todo $x = 0$, y así sucesivamente hasta llegar a la ecuación (2.9), donde el grado de una función racional es $n + m$ y el número de parámetros es $N+1$.

$$f^{(m+n)}(0) = R^{(m+n)}(0) \quad (2.9)$$

Si se expande la función ecuación (2.9) por series de Taylor, para todo $x = 0$, se obtienen las ecuaciones (2.10) y (2.11).

$$F(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m \quad (2.10)$$

$$R(x) = b_0 + b_1x + b_2x^2 + \dots + b_mx^m \quad (2.11)$$

Aplicando la aproximación de Padé, a un retardo de tiempo $F(s) = e^{-\theta s}$, se obtiene la ecuación (2.12).

$$R_m(s) = \frac{a_0 + a_1x + a_2x^2 + \dots + a_mx^m}{b_0 + b_1x + b_2x^2 + \dots + b_mx^m} \quad (2.12)$$

El polinomio de la ecuación (2.12), se puede reemplazar en función a un nuevo polinomio, obteniendo la ecuación (2.13).

$$R_m(s) = \frac{Q_m(-\theta_s)}{Q_m(\theta_s)} \quad (2.13)$$

Para obtener los parámetros de Q_m se utiliza la ecuación (2.14).

$$Q_m(\theta_s) = \sum_{j=0}^m \frac{(m+j)!}{j!(m-j)!} (\theta_s)^{m-j} \quad (2.14)$$

Por medio de la ecuación (2.14), se calcula la aproximación de Padé, donde m declara el orden de la aproximación. En este trabajo se usó la aproximación de primer orden (ecuación (2.15)) y de segundo orden (ecuación (2.16)), ya que son las más comunes en las practicas [1].

$$e^{-\theta s} \approx \frac{1 - \frac{\theta}{2}s}{1 + \frac{\theta}{2}s} \quad (2.15)$$

$$e^{-\theta s} \approx \frac{1 - \frac{\theta}{2}s + \frac{\theta^2}{12}s^2}{1 + \frac{\theta}{2}s + \frac{\theta^2}{12}s^2} \quad (2.16)$$

2.3 PERTURBACIONES

Una perturbación es una señal que afecta el valor en la salida de un sistema, puede generarse tanto dentro del sistema como externa al sistema. Si la perturbación se genera dentro del sistema es llamada interna, mientras que una perturbación externa se genera fuera de la misma e instituye una entrada. Las perturbaciones se asignaron en la entrada y salida de la planta, tal como se muestra en la Figura 2.2.

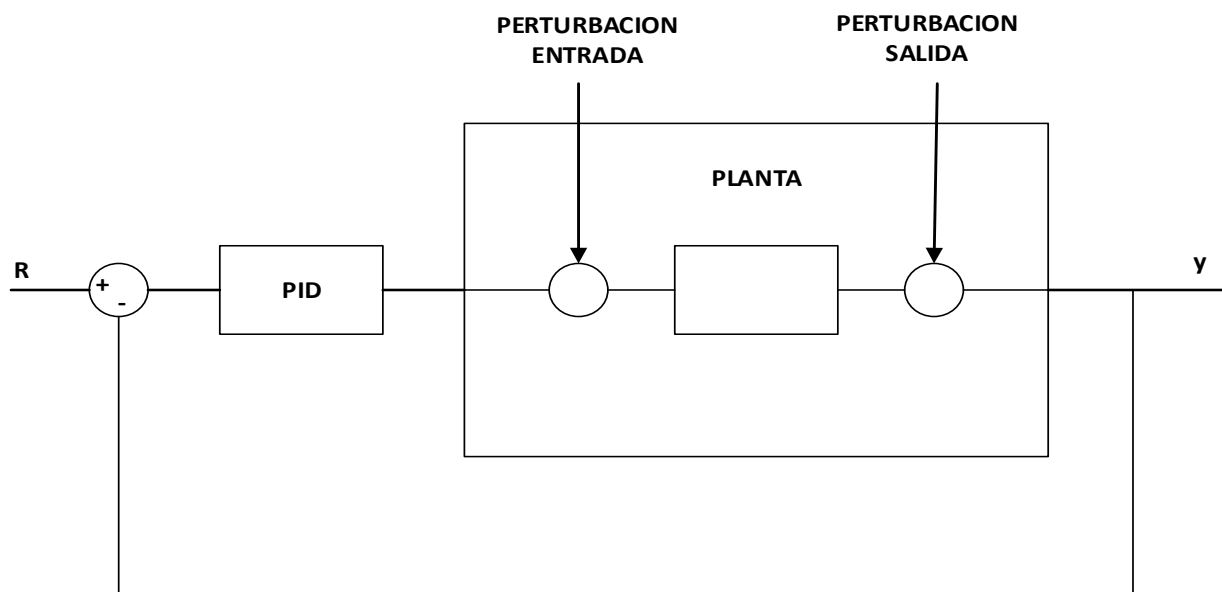


Figura 2.2 Diagrama de bloques de un sistema de control a lazo cerrado con perturbaciones

Las razones por la cual estas perturbaciones se situaron en estos puntos en particular, es para probar la capacidad del controlador ante ellas. Ejemplo: una válvula, que tiene juego o una imperfección, se puede girar un cuarto de ella y no reacciona, eso hace que se degrade la acción de control del PID, ya que por esa imperfección, el controlador actúa, y como no reacciona de inmediato el controlador incrementa la acción de control, otro ejemplo sería un ruido que se sobrepone o encuentra un camino a la salida de la planta, lo que hace que en la salida de la planta llegue una perturbación lo cual hace que la salida de la planta cambie de valor obteniendo algo que no es correcto, pero en realidad la planta no está cambiando, cambia es la salida y esto se debe a la perturbación que en este caso sería el ruido, por el error en la salida de la planta.

2.4 CONTROLADORES PID

Los controladores PID son los más utilizados en la industria en comparación a los demás tipos de controladores que existen, debido a que se usan en aproximadamente el 95% de las operaciones de circuito cerrado de automatización industrial. La versatilidad, simplicidad y capacidad que poseen al momento de solventar los problemas que se obtienen en los procesos dinámicos, son prácticos, ya que corrige el error entre un valor medido y el valor que se quiere obtener.

Los grupos de controladores PID, incluyen tres acciones: proporcional, se encarga de corregir los picos y las oscilaciones de la respuesta del sistema; integral, corrige el error en estado estable aproximándolo a cero y la acción derivativa, se encarga de corregir los cambios rápidos en el transitorio. Por lo cual los controladores de este tipo son denominados P, I, PI, PD y PID.

La ecuación (2.17) pertenece a un controlador PID en tiempo continuo.

$$u(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}] \quad (2.17)$$

2.5 SINTONIZACION DE CONTROLADORES PID

La sintonización radica en determinar los parámetros del controlador PID (K_p , T_i , T_d) para así estimar una respuesta aceptable del mismo, que permita expresarlo en un modelo matemático.

El valor de K_p se conoce como ganancia del controlador, aunque muchos controladores utilizan banda proporcional en lugar de la ganancia. La banda proporcional se define como el cambio en la variable de entrada que provoca una variación del 100% de la salida [2].

Para calcular el valor de la banda proporcional se utiliza la ecuación (2.18).

$$BP = \frac{100 \Delta M}{K_p * E} \quad (2.18)$$

Donde:

- BP = Banda Proporcional.
- E = Alcance.
- ΔM = Variación Máxima de Salida.

En esta tesis no se utilizará banda proporcional, ni la ecuación (2.18), solo se hizo referencia de otra forma de utilizar la ganancia de un controlador.

Anti Reset – wind up: el modo integral del controlador tiene la característica de saturarse en aquellos casos en que el error persiste a lo largo de un tiempo prolongado. Ejemplo: en un controlador neumático, este fenómeno de evidencia en el fuelle integral, que se infla hasta su límite físico. Por esta característica se le conoce como “*anti – reset wind up*” que matemáticamente se puede interpretar como un gran incremento de la integral del error, que se produce aun cuando el error es pequeño, debido a su persistencia en el tiempo. Para que el término del modo integral en el algoritmo disminuya su valor, resulta necesario que el error invierta su signo, y persista así hasta eliminar la saturación. Por tal motivo, pueden producirse apreciables sobre picos en la respuesta del sistema. Este efecto se magnifica en sistemas con retardo o tiempo muerto, o en aquellos en los que el error tiene el mismo signo la mayor parte del tiempo.

El “*anti – reset wind up*” elimina este efecto limitando el incremento del término integral del algoritmo de control [3].

Para el ajuste de parámetros del regulador se realizan de dos formas:

Empíricamente: este sistema puede llegar a ser muy lento en caso de que el tiempo de respuesta sea grande y para evitar este tipo de problema, se utiliza los métodos de Ziegler-Nichols, los cuales se toman en base a unas medidas observadas en la respuesta del sistema y arrojan valores teóricos de los parámetros del controlador. Valores que se adquieren como referencia y se les realiza un mejor ajuste a partir de ellos [4].

Teóricamente: para lo que se determinan de forma analítica los valores del regulador. Se requiere el conocimiento expreso de la función de transferencia del proceso [4].

Los métodos a utilizar para la sintonización empírica de controladores PID en este trabajo de grado son: Ziegler-Nichols y Dahlin.

2.5.1 Método de Ziegler-Nichols a lazo abierto

Son dos métodos clásicos de ajuste empírico de los parámetros de un controlador PID de los cuales se utilizará uno solo de estos dos métodos para la sintonización. Estos métodos son ampliamente utilizados, bien en su forma original o con versiones mejoradas [4].

Método de repuesta ante un escalón: método de respuesta de la planta ante un escalón unitario como se muestra en la Figura 2.3, se obtiene de forma experimental y si la respuesta no tiene oscilaciones y presenta forma de “S”, puede obtenerse los valores del controlador PID al aplicarse este método. El procedimiento se basa en aplicar una recta tangente en el punto de inflexión de la curva en forma de S, para así obtener el valor de los parámetros K, L y T y con estos parámetros, se obtiene la sintonización del controlador por medio de este método [5].



Figura 2.3 Respuesta ante un escalón.

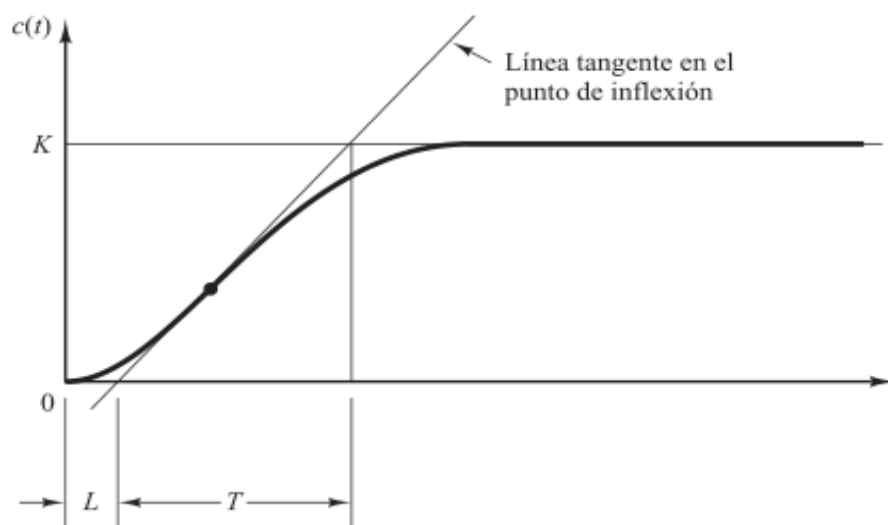


Figura 2.4 Respuesta de un escalón unitario para sintonización del controlador.

Se puede observar en la Figura 2.4; que L es el tiempo muerto, T es la constante de tiempo del sistema y K es la ganancia para un escalón unitario, para realizar la sintonización por este método, se utiliza la Tabla 2.1.

Tabla 2.1 Parámetros de sintonización por el método de Ziegler-Nichols a lazo abierto.

Tipo de Controlador	Ganancia Proporcional	Tiempo de Integración	Tiempo Derivativo
P	$\frac{1}{K} \left(\frac{L}{T}\right)^{-1}$	∞	0
PI	$\frac{0.9}{K} \left(\frac{L}{T}\right)^{-1}$	$3.33L$	0
PID	$\frac{1.2}{K} \left(\frac{L}{T}\right)^{-1}$	$2L$	$\frac{1}{2}L$

2.5.2 Método de Dahlin a lazo abierto

El procedimiento para este método es muy parecido al de Ziegler-Nichols. Los parámetros de sintonización se obtienen de la Figura 2.4 y sus ecuaciones para el cálculo de la sintonización se calculan a partir de las ecuaciones en la Tabla 2.2.

Tabla 2.2 Parámetros de sintonización por el método de Dahlin a lazo abierto.

Controlador	Parámetro de ajuste	Ecuación
PID	K_p	$\frac{1.2}{2K} \left(\frac{t_0}{\tau}\right)^{-1}$
	τ_i	τ
	τ_d	$\frac{\bar{t}_0}{2}$

2.6 TRANSFORMADA Z

La transformada Z es una herramienta operacional que se utiliza para los sistemas en tiempo discreto (complejo), semejante a ella, la transformada de Laplace, que se usa para sistemas de control en tiempo continuo. Al igual que la transformada de Laplace la transformada Z tiene una tabla de sus principales transformada como se muestra en la Tabla 2.3.

Tabla 2.3 Algunas transformadas Z

	$X(s)$	$x(t)$	$x(kT)$ o $x(k)$	$X(z)$
1.	—	—	Delta de Kronecker $\delta_0(k)$ 1, $k = 0$ 0, $k \neq 0$	1
2.	—	—	$\delta_0(n - k)$ 1, $n = k$ 0, $n \neq k$	z^{-k}
3.	$\frac{1}{s}$	$1(t)$	$1(k)$	$\frac{1}{1 - z^{-1}}$
4.	$\frac{1}{s + a}$	e^{-at}	e^{-akT}	$\frac{1}{1 - e^{-aT}z^{-1}}$
5.	$\frac{1}{s^2}$	t	kT	$\frac{Tz^{-1}}{(1 - z^{-1})^2}$
6.	$\frac{2}{s^3}$	t^2	$(kT)^2$	$\frac{T^2 z^{-1}(1 + z^{-1})}{(1 - z^{-1})^3}$
7.	$\frac{6}{s^4}$	t^3	$(kT)^3$	$\frac{T^3 z^{-1}(1 + 4z^{-1} + z^{-2})}{(1 - z^{-1})^4}$
8.	$\frac{a}{s(s + a)}$	$1 - e^{-at}$	$1 - e^{-akT}$	$\frac{(1 - e^{-aT})z^{-1}}{(1 - z^{-1})(1 - e^{-aT}z^{-1})}$
9.	$\frac{b - a}{(s + a)(s + b)}$	$e^{-at} - e^{-bt}$	$e^{-akT} - e^{-bkT}$	$\frac{(e^{-aT} - e^{-bT})z^{-1}}{(1 - e^{-aT}z^{-1})(1 - e^{-bT}z^{-1})}$
10.	$\frac{1}{(s + a)^2}$	te^{-at}	kTe^{-akT}	$\frac{Te^{-aT}z^{-1}}{(1 - e^{-aT}z^{-1})^2}$
11.	$\frac{s}{(s + a)^2}$	$(1 - at)e^{-at}$	$(1 - akT)e^{-akT}$	$\frac{1 - (1 + aT)e^{-aT}z^{-1}}{(1 - e^{-aT}z^{-1})^2}$

De una función en el tiempo $x(t)$, siendo t positivo, o de los valores en secuencia de $x(kt)$, en donde k adopta valores de cero o enteros positivos y T vendría siendo el periodo de muestreo, se obtiene la ecuación (2.19).

$$X(z) = Z[x(t)] = Z[x(kT)] = \sum_{k=0}^{\infty} x(kT)z^{-k} \quad (2.19)$$

Si la señal es una secuencia de números para $x(k)$, entonces la transformada z de la ecuación (2.19) sería:

$$X(z) = Z[x(k)] = \sum_{k=0}^{\infty} x(k)z^{-k} \quad (2.20)$$

La ecuación (2.20) se le conoce con el nombre de transformada z unilateral, y en ella se supone que $x(k) = 0$, para todo $k < 0$ o $x(t) = 0$, para $t < 0$. Transformando de esta manera la secuencia en el tiempo $x(kt)$ en una función compleja llamada $x(z)$ [6].

La transformada Z bilateral tiene las mismas ecuaciones, ecuación (2.19) y ecuación (2.20), la diferencia es que para este tipo de transformada bilateral no implica un periodo de muestreo; en el tiempo $x(t)$ implícitamente, no incluye T, por ende, se define la transformada z de $x(t)$ para $-\infty < t < \infty$, o de $x(k)$ donde k adopta valores enteros como $\pm 1, \pm 2, \pm 3, \pm \infty$. Este tipo de transformada bilateral asume que para la función en el tiempo $x(t)$ es distinta de cero para todo $t < 0$ y para la secuencia de valores de $x(k)$ es distinta de cero para $k < 0$.

El tipo de transformada Z que se usará para este trabajo es la unilateral debido a presenta la solución más apropiada para el desarrollo y análisis de diferentes tipos de sistemas [6].

2.6.1 Métodos de la transformada Z

Existen seis métodos para realizar la transformada Z los cuales son:

1. Aproximación hacia adelante o método de Euler I.
2. Aproximación hacia Atrás o método de Euler II.
3. Aproximación Bilineal o método Bilineal.
4. Aproximación Mapeo Polo Cero.
5. Aproximación Invariante al Impulso.
6. Aproximación Invariante al Escalón.

Estos métodos presentan variación de precisión al aplicarse cada uno de ellos, en la Figura 2.5 se muestra la precisión y complejidad matemática que posee.

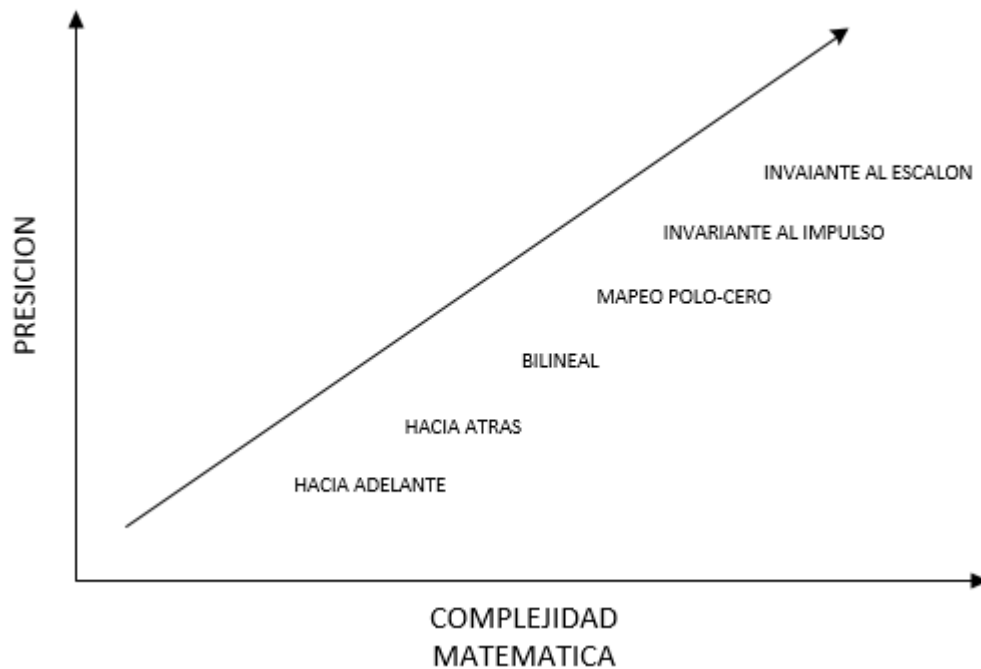


Figura 2.5 Gráfica de precisión y complejidad de los métodos de la transformada Z.

El método que se utilizó es el Invariante al Escalón, ya que como los esfuerzos de control van enfocados principalmente en seguimiento y regulación, los métodos son ajustados para obtener una cercanía con la respuesta al escalón y también porque este método es el más preciso de todos.

Aproximación Invariante al Escalón: para este método se parte de la función en continua de $G(s)$ y se multiplicara por una integral como se muestra en la ecuación (2.21)

$$X(z) = Z[x(k)] = \sum_{k=0}^{\infty} x(k)z^{-k} \quad (2.21)$$

Posteriormente, realizando el uso de la tabla de transformada de Laplace a transformada z, se obtiene $G_e(z)$, que, al ser multiplicada por una ganancia, quedaría como una integral discreta como se muestra en la ecuación (2.22).

$$AG_e(z) = \frac{z}{z-1}G(z) \quad (2.22)$$

$$G(z) = AG_e(z) \frac{z-1}{z} \quad (2.23)$$

Quedando así la ecuación (2.23), la cual es la que se usa para discretizar de la manera más precisa. El valor de la ganancia se obtiene por el teorema del valor final a través de la ecuación (2.24).

$$\lim_{z \rightarrow 1} G(z) = \lim_{s \rightarrow 0} G(s) \quad (2.24)$$

2.7 CONTROLADOR PID DISCRETIZADO

En esta entrada se realizará el estudio de cómo obtener la ecuación de un controlador PID discreto a partir de la ecuación diferencial (2.17). El término derivativo de esta expresión se puede sustituir por la diferencia de primer orden y el término integral se aproximará por el primer método de Euler I o aproximación hacia adelante, quedando la ecuación (2.25) [4].

$$u(k) = K[e(k) + \frac{T}{Ti} \sum_{i=0}^{k-1} e(i) + \frac{Td}{T}(e(k) - e(k-1))] \quad (2.25)$$

Para que la expresión de la ecuación (2.25) evite el sumatorio, se obtendrá la expresión para $(k-1)$ obteniendo la ecuación (2.26).

$$u(k-1) = K[e(k-1) + \frac{T}{Ti} \sum_{i=0}^{k-2} e(i) + \frac{Td}{T}(e(k-1) - e(k-2))] \quad (2.26)$$

Restando de la ecuación (2.25) la ecuación (2.26) se obtiene la expresión del controlador PID discreto, ecuación (2.27). De la ecuación (2.27) se puede observar los valores que toman q_0 (ecuación (2.28)), q_1 (ecuación (2.29)) y q_2 (ecuación (2.30)).

$$u(k) = u(k-1) + [q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)] \quad (2.27)$$

$$q_0 = K \left(1 + \frac{Td}{T} \right) \quad (2.28)$$

$$q_1 = -K \left(1 + 2 \frac{Td}{T} - \frac{T}{Ti} \right) \quad (2.29)$$

$$q_2 = K \left(\frac{Td}{T} \right) \quad (2.30)$$

La función de transferencia en Z quedaría de la siguiente forma, ecuación (2.31)

$$G(z) = \frac{U(z)}{E(z)} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad (2.31)$$

www.bdigital.ula.ve

CAPITULO 3

DISCRETIZACION DE LOS SISTEMAS

En este capítulo se explica cómo se discretizaron las ecuaciones de los diferentes tipos de plantas y el controlador PID, se comprueba su funcionabilidad utilizando la herramienta de Matlab Simulink y Wolfram Mathematica 10.4.

3.1 DISCRETIZACION DE LAS PLANTAS

Como se menciona en el capítulo 2, se utilizaron cuatro tipos de plantas (integrante, primer orden, primer orden integrante y segundo orden), el método que se utiliza para discretizar es el invariante al escalón, pero para verificar la efectividad del método, se usó y se comparó con el programa Wolfram Mathematica 10.4, se pudo apreciar que el resultado obtenido manualmente coincide con el resultado del programa. A continuación, se muestra la validación de la discretización de cada una de las plantas con las curvas de cada una de ellas, estas curvas deben superponerse entre sí para garantizar el cumplimiento de la discretización con respecto a las ecuaciones en el dominio del tiempo.

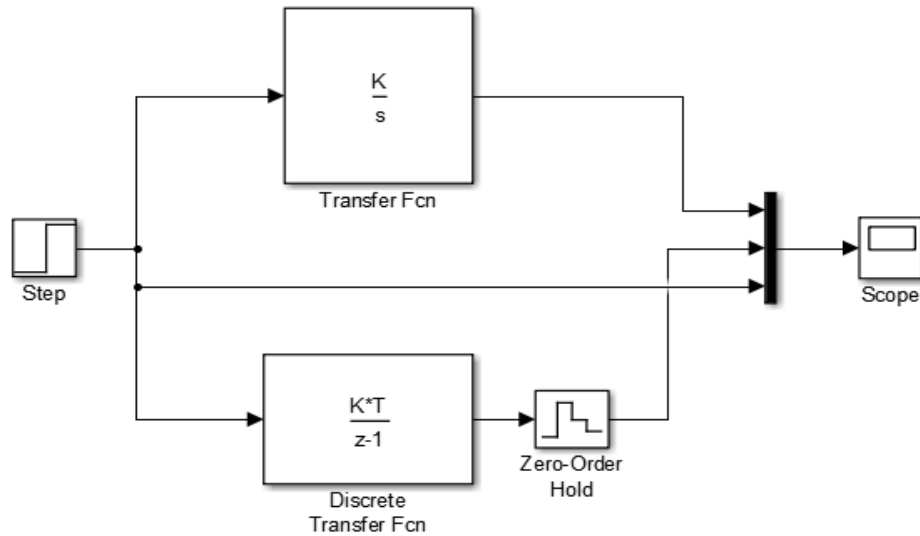


Figura 3.1 Comparación de las respuestas en tiempo continuo y discreto de la planta integrante.

Como se muestra en la Figura 3.1, la cual representa la planta integrante en tiempo continuo y discreto, para así, a través de la herramienta Simulink Matlab se comprueba que la respuesta de ambos sistemas coincide.

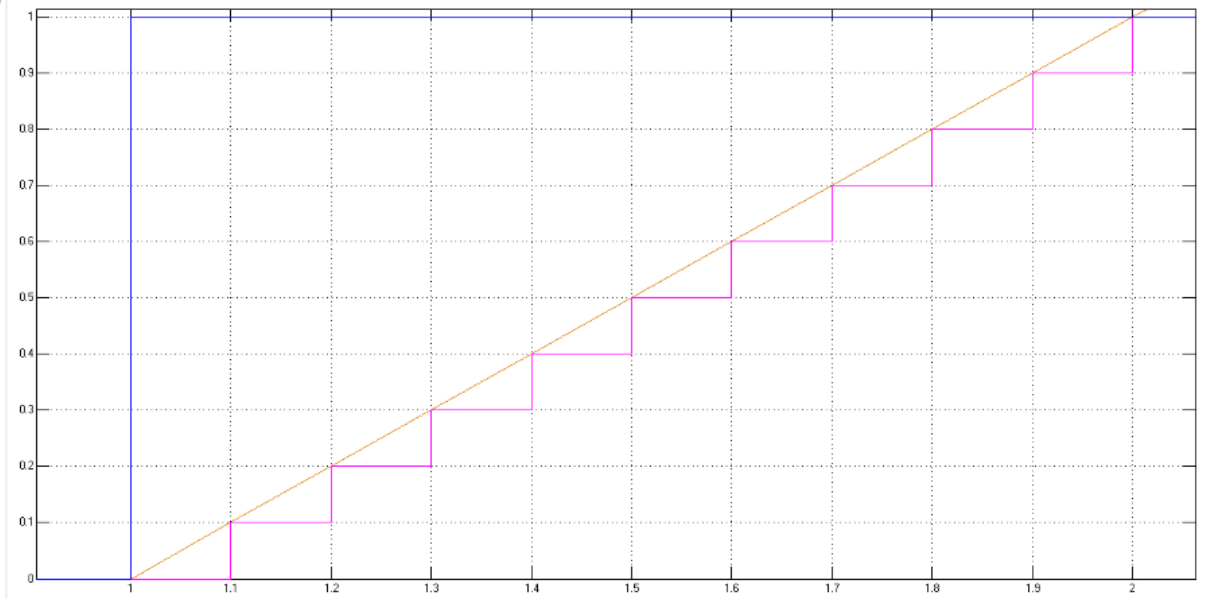


Figura 3.2 Respuesta de la planta integrante en tiempo continuo y discreto.

En la Figura 3.2 se muestra cómo se superpone ambas curvas, mostrando la veracidad del sistema para la planta.

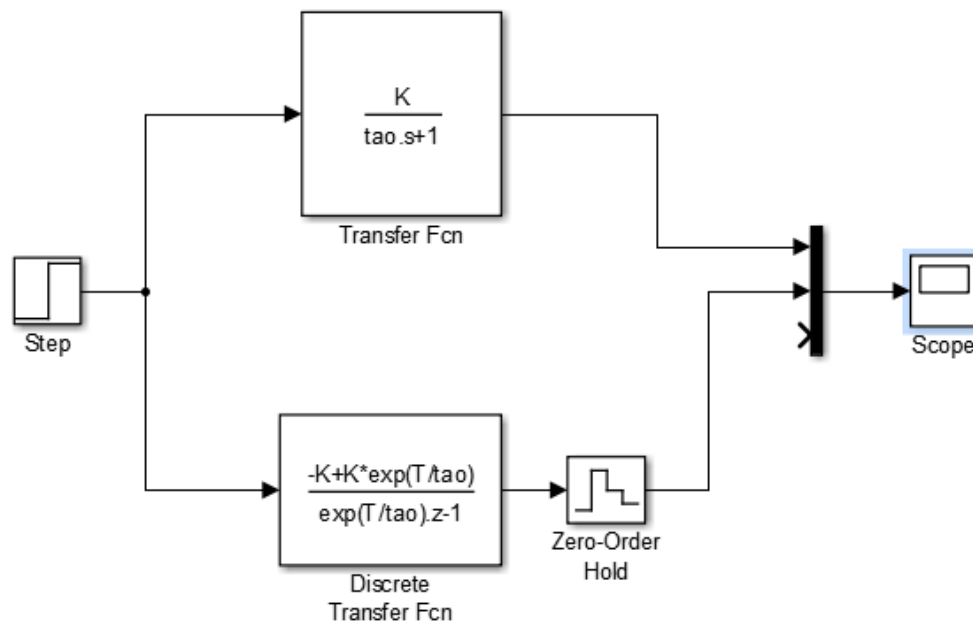


Figura 3.3 Comparación de las respuestas en tiempo continuo y discreto. Planta de primer orden.

La Figura 3.3 muestra la representación de la planta de primer orden para la verificación de que ambos sistemas concuerden entre sí, tanto el discreto como el continuo.

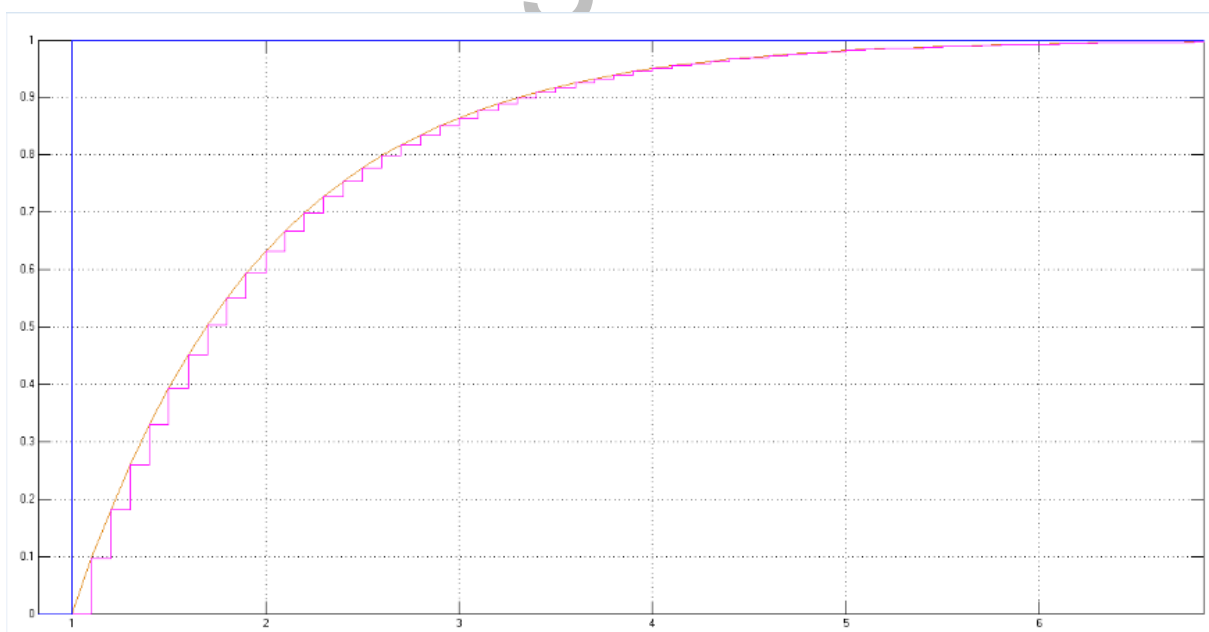


Figura 3.4 Respuesta de la planta de primer orden en tiempo continuo y discreto.

Tal y como se muestra en la Figura 3.4, ambos sistemas se superponen, por lo tanto, la planta de orden 1 cumple con lo requerido.

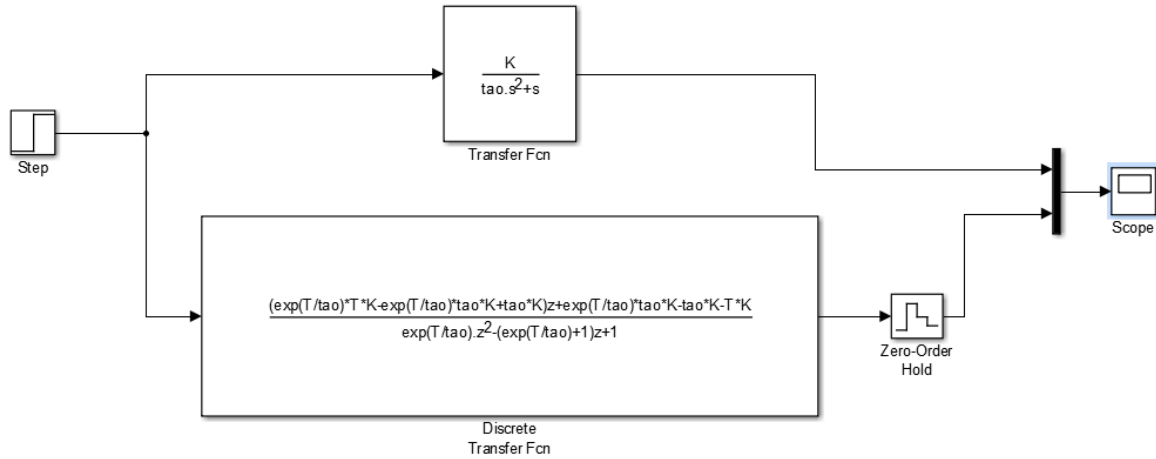


Figura 3.5 Comparación de respuesta de la planta de primer orden integrante en tiempo continuo y discreto.

La Figura 3.5 es la forma de la planta de primer orden con un sistema integrante, para esta planta también se prueba la estabilidad de ambos sistemas.

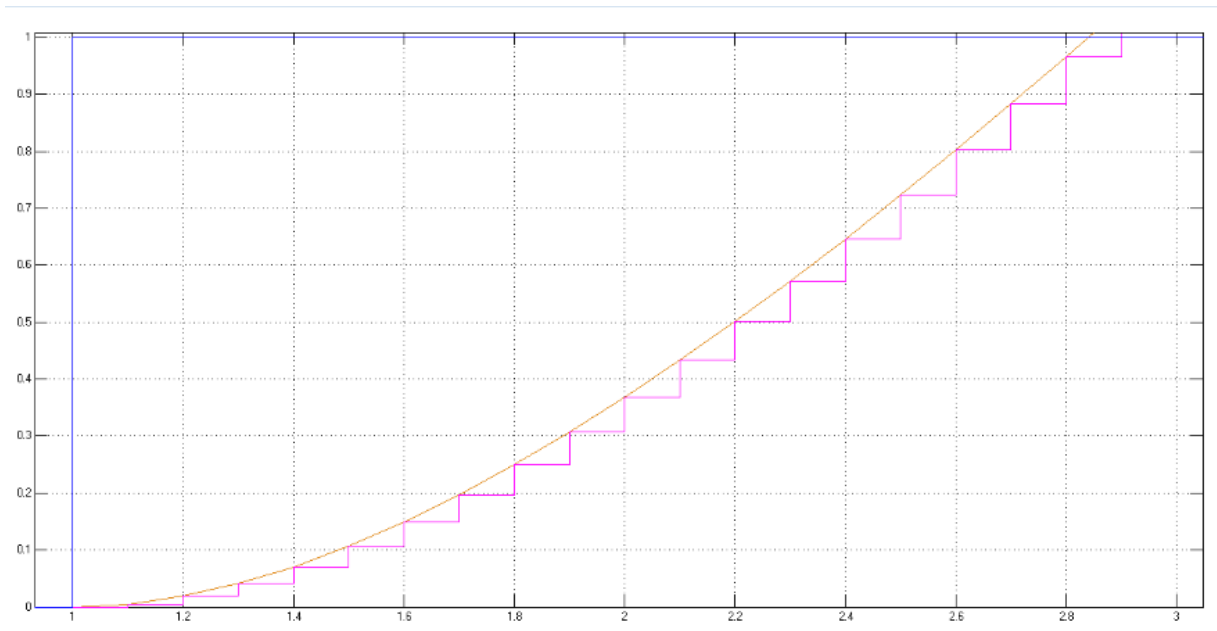


Figura 3.6 Respuesta de la planta de primer orden integrante en tiempo continuo y discreto.

Como se muestra en Figura 3.6, las curvas de ambos sistemas se superponen, lo cual garantiza la funcionalidad del mismo.

Para la planta de segundo orden, se realizó el mismo procedimiento, tal y como se muestra en la Figura 3.7.

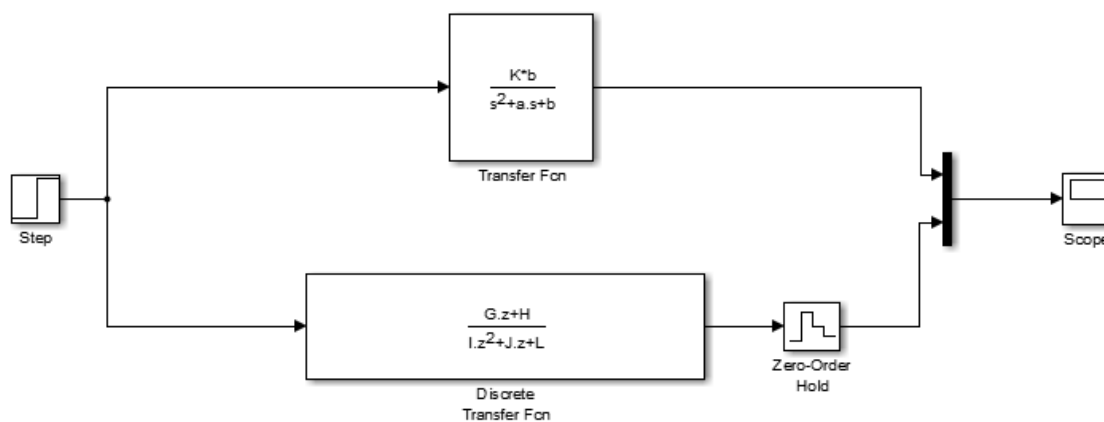


Figura 3.7 Comparación en tiempo continuo y discreto de la planta de segundo orden.

Como se muestra en las figuras anteriores, para esta planta, también se puede apreciar que ambos sistemas cumplen y coinciden entre sí, para su comprobación, tal y como se observa en la Figura 3.8.

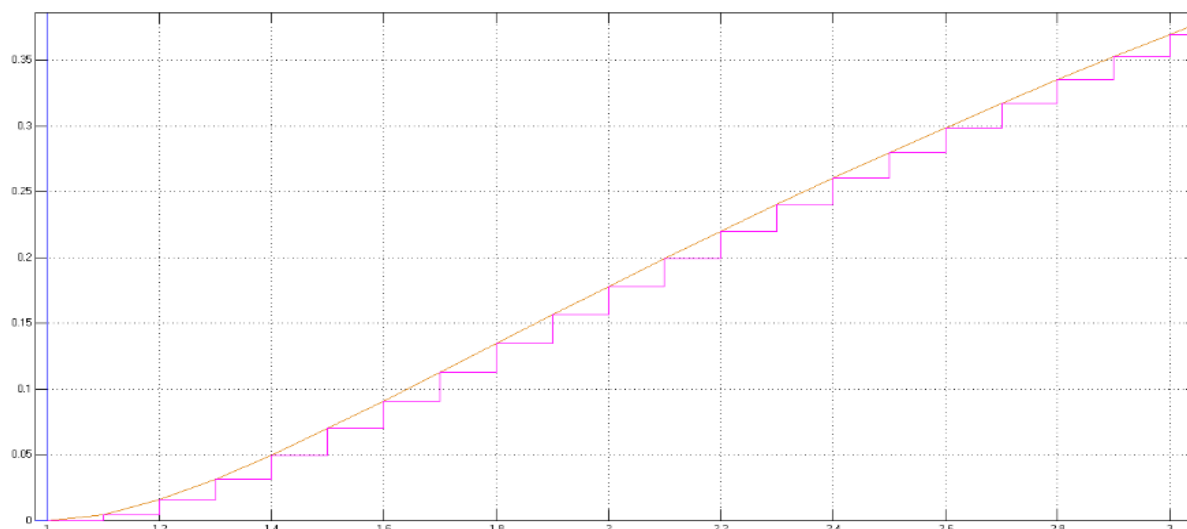


Figura 3.8 Respuesta de la planta de segundo orden en tiempo continuo y discreto.

3.1.1 Plantas con tiempo muerto

Obtenidas las plantas discretizadas, ahora se le agrega el tiempo muerto y este mismo se lleva al dominio complejo o discreto. El método que se empleó para la aplicar el tiempo muerto fue a través del método de Padé, aunque este método es el más utilizado, al momento de discretizar el tiempo muerto se pudo observar que las curvas o respuestas no coincidían con lo esperado, por lo tanto, se recurrió a utilizar desplazamientos con líneas de retardo, para desplazar dicho tiempo, de esta manera se pudo compensar el error de utilizar aproximaciones.

A continuación, se mostrarán las gráficas que representan a cada una de las plantas con el tiempo muerto agregado, para esto se utilizó Matlab Simulink, para demostrar la validación de estas plantas al momento de sumarle el tiempo muerto.

En la Figura 3.9 se muestra el diagrama de la planta con el tiempo muerto en continuo y discreto, como antes se menciona, la discretización de este tiempo no funciona, por ende, también se ve reflejado el icono que realiza el desplazamiento. Este diagrama es prácticamente el mismo para todas las plantas que se utilizan en este trabajo, lo único que cambia son las ecuaciones de las plantas. El tiempo muerto que se utiliza es de segundo para todas y se mostrará la diferencia de cada una de las curvas más adelante.

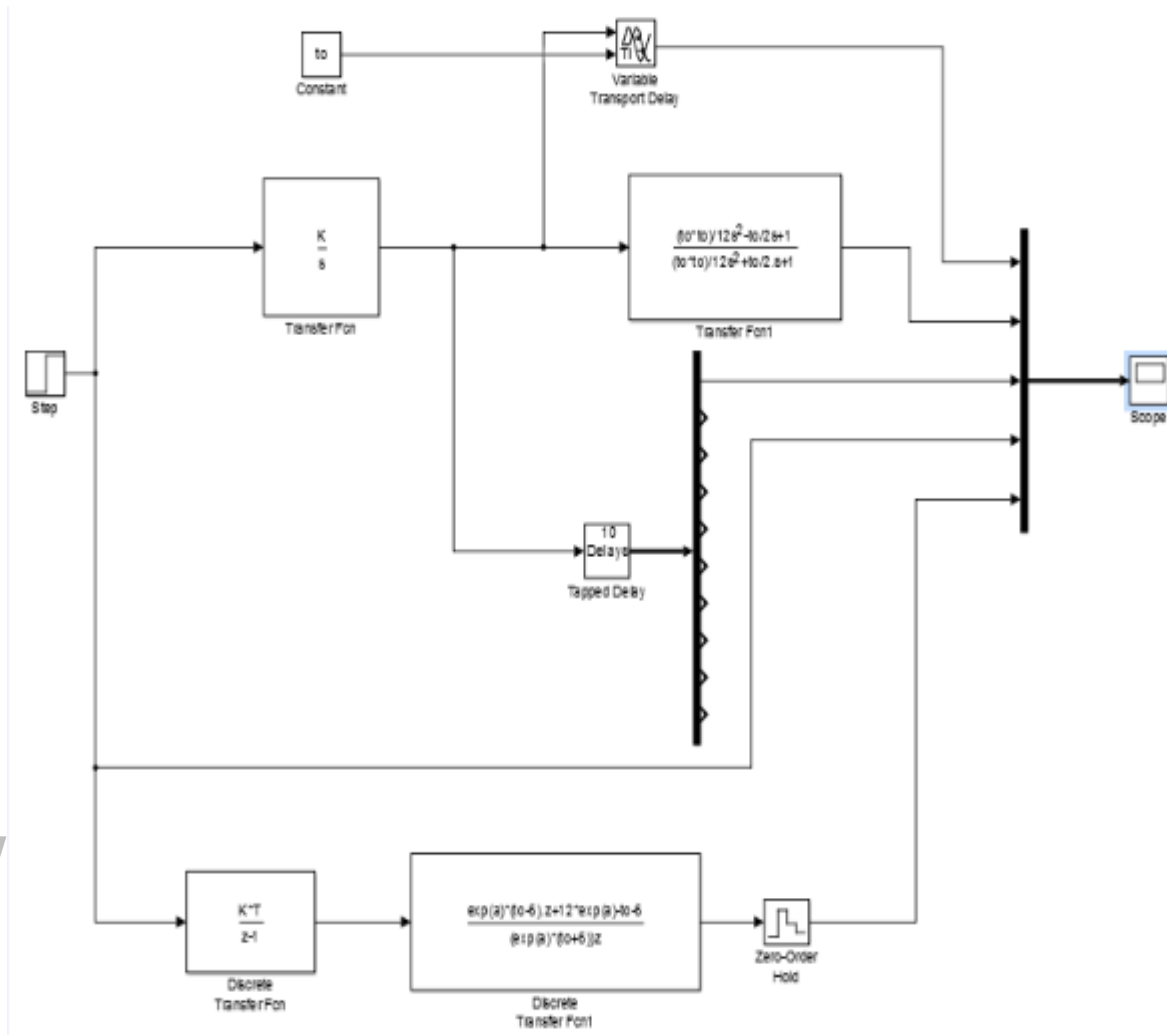


Figura 3.9 Comparación de la planta integradora con tiempo muerto de segundo orden.

La Figura 3.10 muestra las diferentes curvas que produce el diagrama de la Figura 3.9, cada una de ellas tiene su representación, la línea roja es el escalón, la amarilla es el tiempo muerto real, la morada es el tiempo muerto de segundo orden en continua, la azul es el tiempo muerto con líneas de retardo y la línea verde sería el tiempo muerto discreto de orden dos. Como se muestra en esta gráfica el tiempo muerto discreto no se superpone en ningún momento con la planta, por lo tanto, no es viable esta respuesta, mientras que con el desplazamiento o líneas de retardo si coincide con el funcionamiento, la línea del tiempo muerto en continuo, muestra una variación al principio de la gráfica, esto se debe a que posee fase no mínima y esto ocurre por la aproximación de Padé.

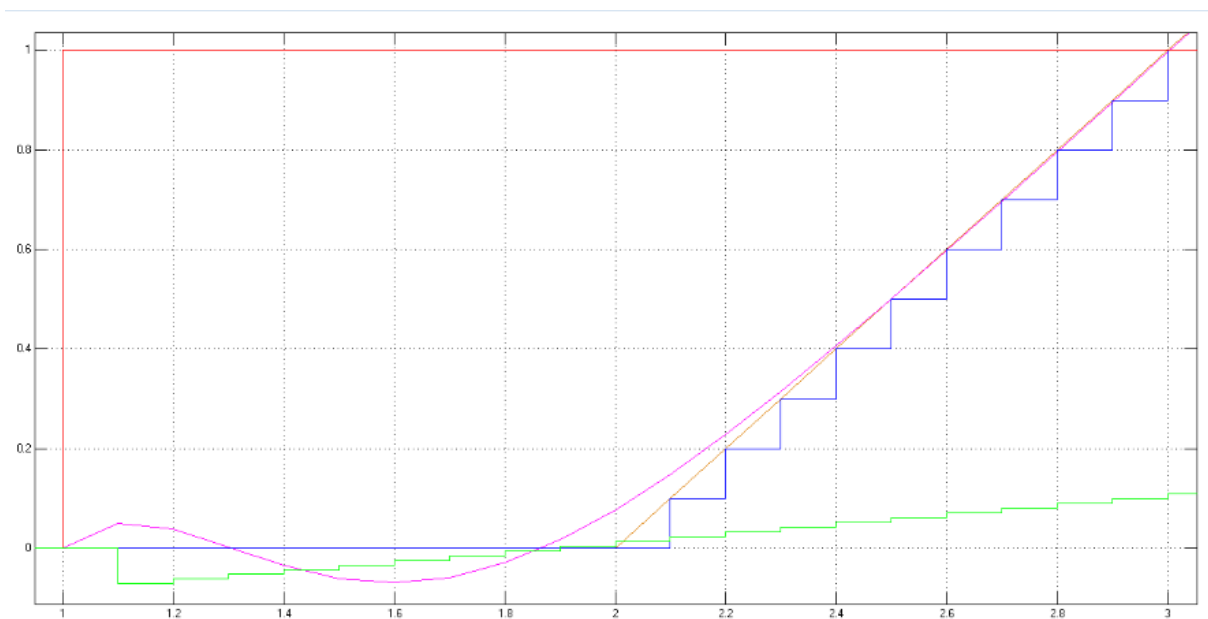


Figura 3.10 Respuesta de la planta integrante con tiempo muerto.

La gráfica de la planta de primer orden más tiempo muerto se analiza en la Figura 3.11 y como se muestra, la curva del tiempo muerto en discreto es similar a la que se muestra en la Figura 3.10, por lo que se puede llegar al mismo desenlace anterior, así como también la curva del tiempo muerto en continuo posee fase no mínima, debido a estas razones se excluyó este método y se utiliza el desplazamiento con líneas de retardo, ya que por aproximación de Padé podría perjudicar al controlador más adelante.

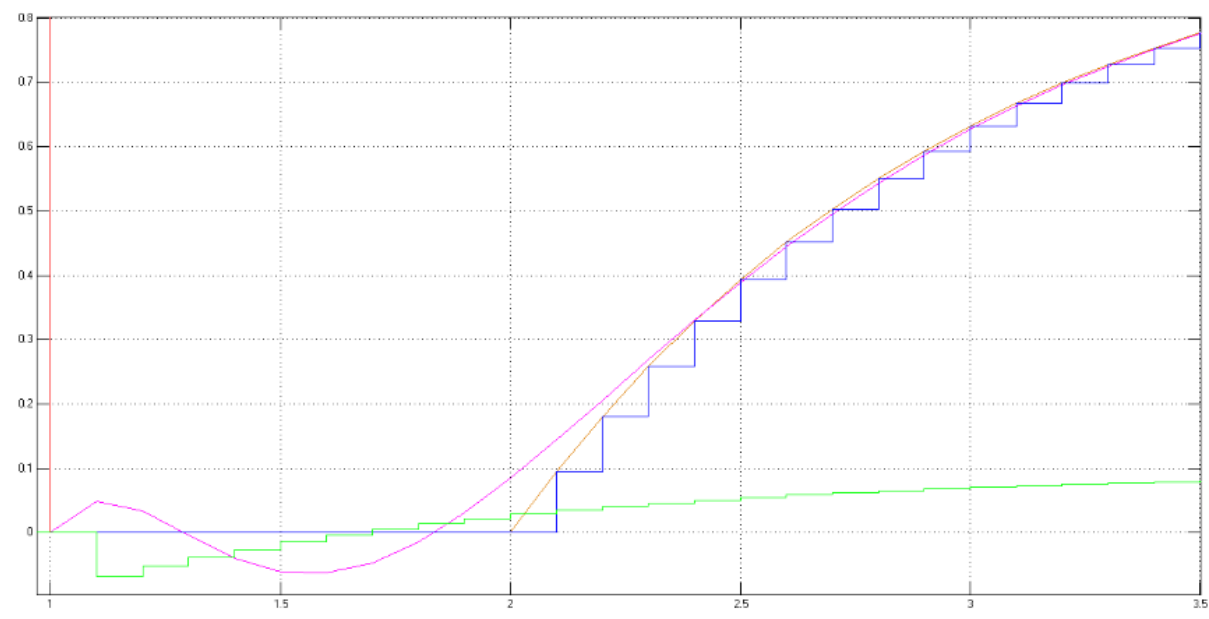


Figura 3.11 Respuesta de la planta de primer orden con tiempo muerto.

La forma de onda del tiempo muerto en discreto no tiende a superponerse a la curva de la planta, lo cual implica una similitud con las gráficas anteriores Figura 3.10 y Figura 3.11, estos efectos se pueden observar en la Figura 3.12.

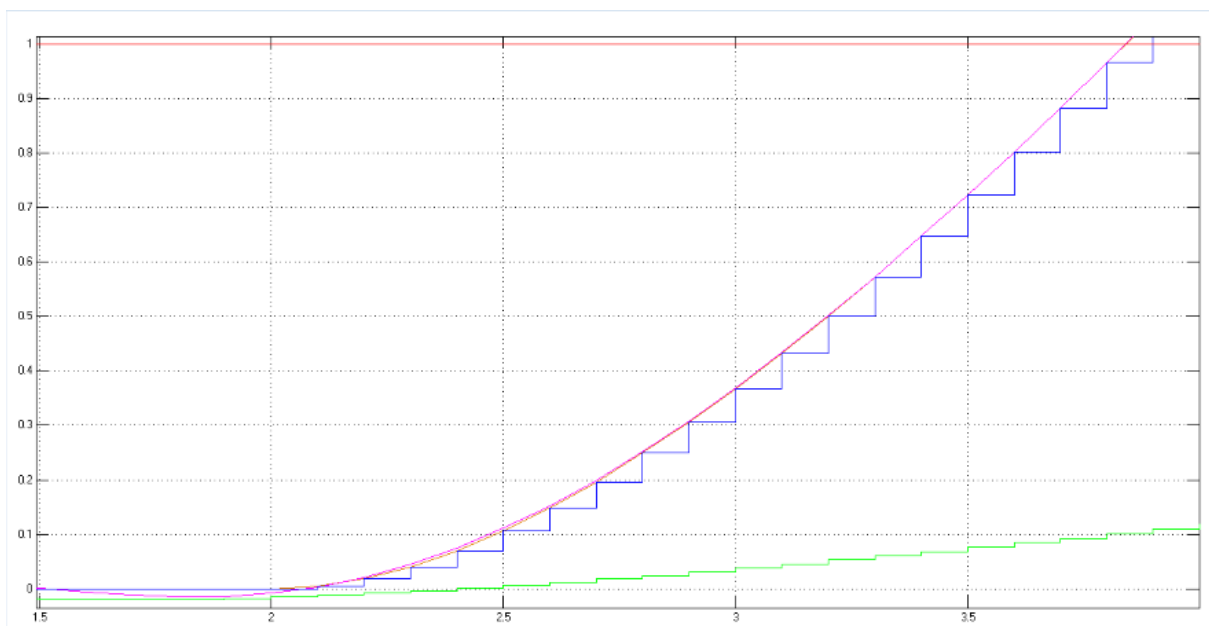


Figura 3.12 Respuesta de la planta de primer orden integrante con tiempo muerto.

En la Figura 3.13 se puede diferenciar de las demás gráficas, ya que se demuestra que el tiempo muerto en discreto tiende a coincidir a las curva de reacción de la planta, pero aun así, no es suficiente, ya que no se superpone a la misma, por lo tanto se puede dar por demostrado que la mejor opción es a través del desplazamiento de líneas de retardo que utilizar el método de aproximación por Padé.

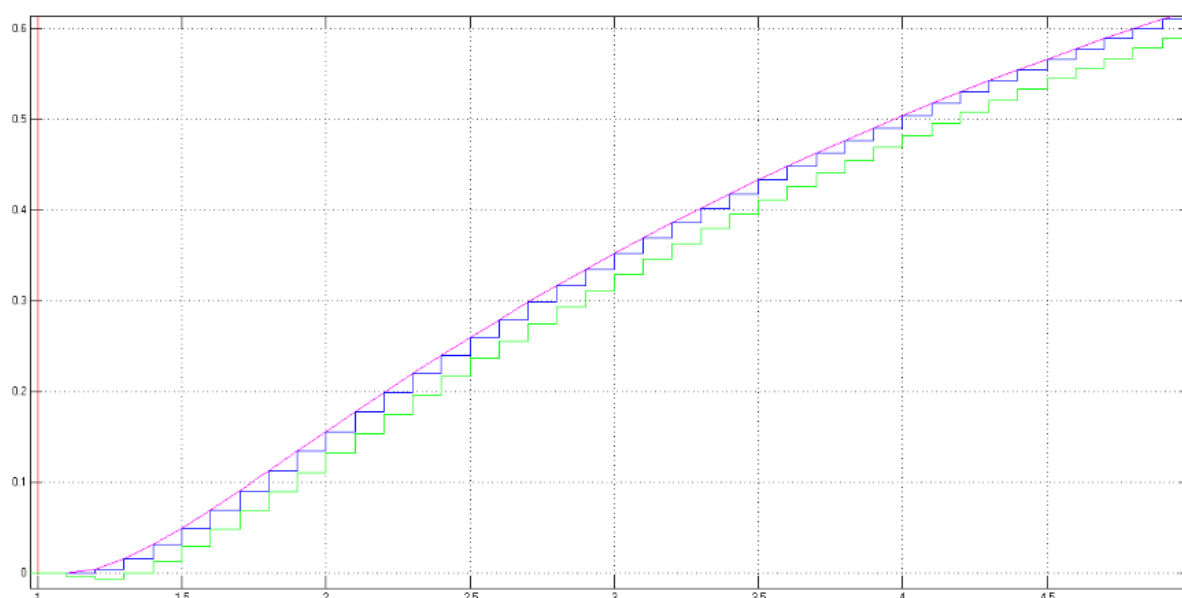


Figura 3.13 Respuesta de la planta de segundo orden con tiempo muerto.

En las siguientes graficas se muestra otra manera de ver la razón por la cual se descartó el método de aproximación por Padé en continuo y discreto,

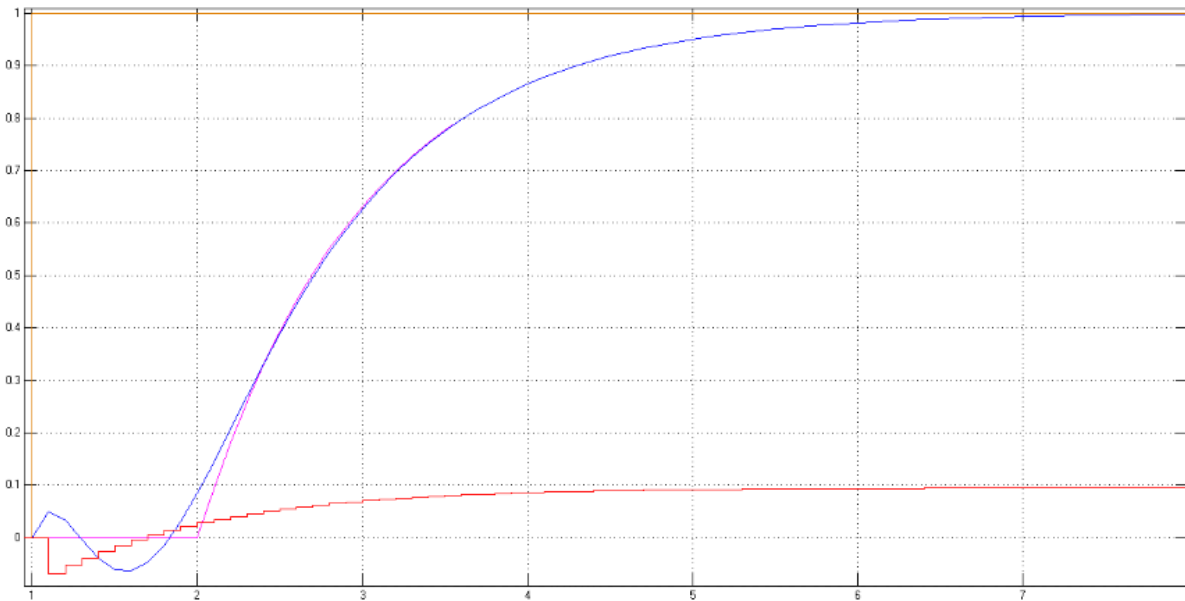


Figura 3.14 Respuesta por el método de aproximación de Padé.

En la Figura 3.14, las formas de ondas que se descartaron para utilizar en este proyecto, la razón está a simple vista, la curva de color azul es la respuesta de la aproximación en tiempo continuo, y como se puede observar, tiene fase no mínima y por esta razón no debería utilizarse, ya que podría perjudicar el sistema más adelante. También, se tiene la curva de respuesta de la aproximación en discreto y como antes se menciona, al momento de discretizar este tiempo muerto, la forma de onda no intenta superponerse en ningún momento, por ende, se descarta. Ahora, al utilizar desplazamiento por líneas de retardo, se manifiesta la respuesta que se quiere obtener, tal y como se muestra en la Figura 3.15.

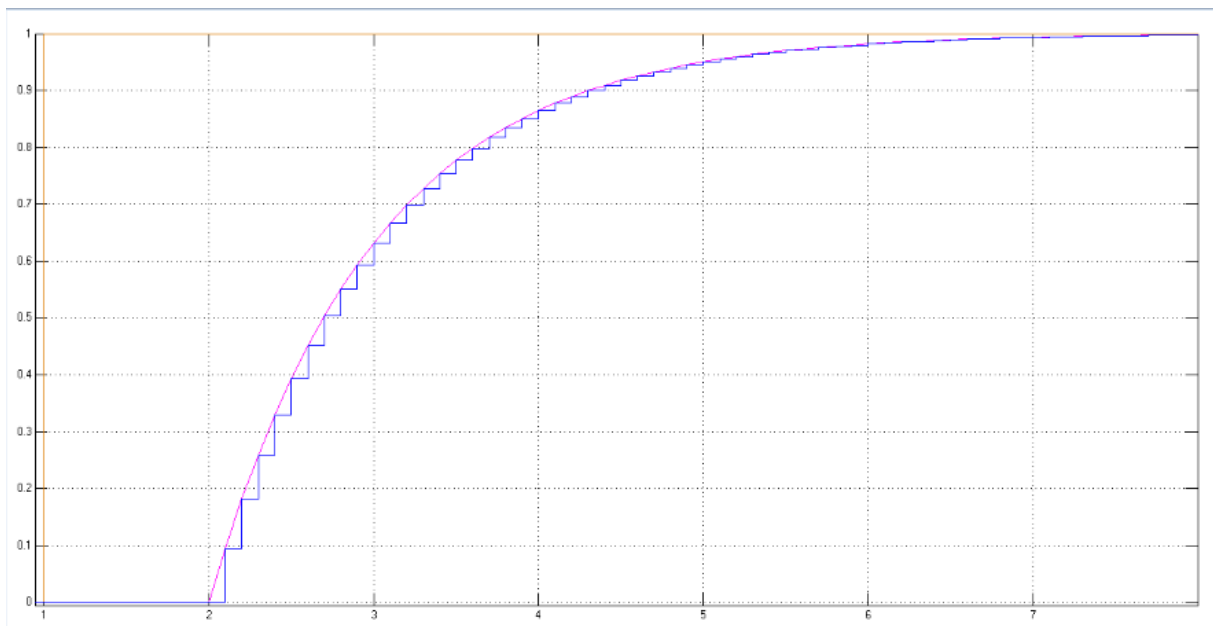


Figura 3.15 Respuesta de tiempo muerto por desplazamiento de líneas de retardo.

3.1.2 Controlador PID

En el capítulo anterior, se mencionó la teoría de controladores PID, como discretizar y los métodos de sintonización, ahora se aplicará este tipo de controlador a los sistemas para realizar las comparaciones de su funcionamiento, tanto en tiempo continuo como en discreto.

3.1.3 Controlador PID en tiempo continuo

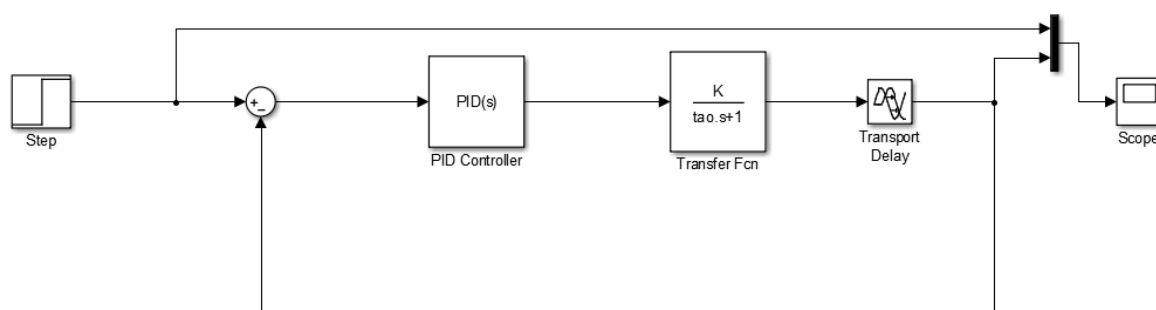


Figura 3.16 Diagrama del controlador PID a lazo cerrado.

La Figura 3.16 muestra el diagrama a lazo cerrado de un controlador PID con una planta de orden uno en tiempo continuo, este diseño es el mismo para los sistemas de todas las plantas que se utilizan en este proyecto.

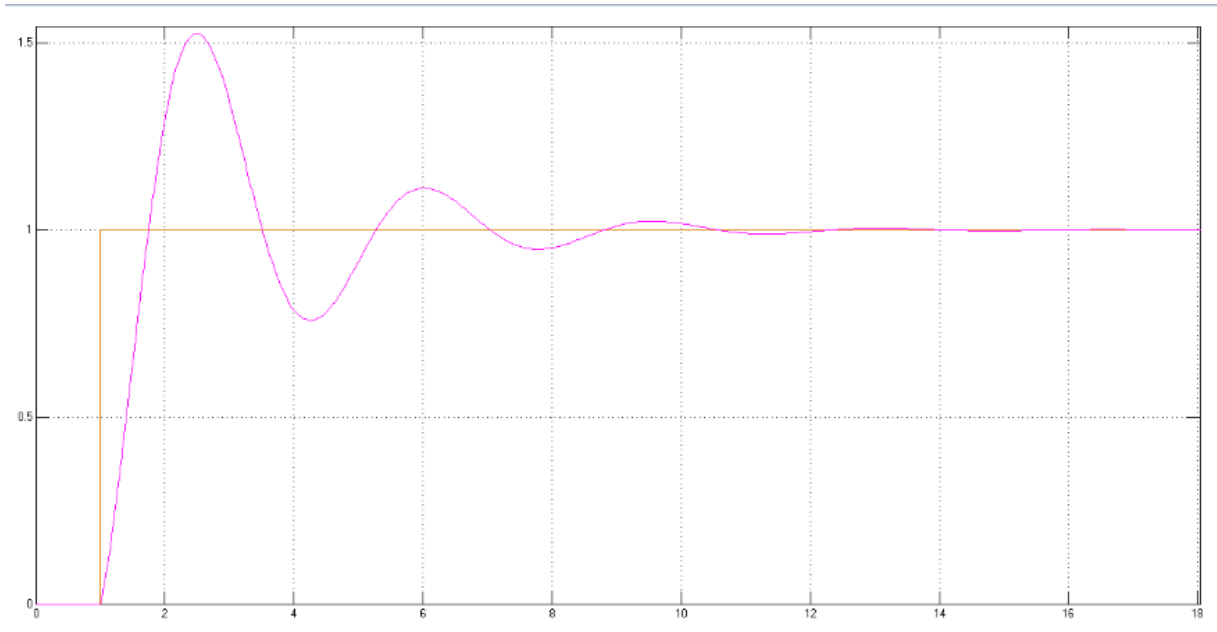


Figura 3.17 Respuesta a lazo cerrado del controlador para una planta integrante.

Como se muestra en la Figura 3.17, el controlador calcula la salida para que la planta siga la referencia, la respuesta del controlador es más agresiva para este tipo de planta (integrante), los parámetros del controlador se calcularon a través de una opción de sintonización que posee Simulink Matlab.

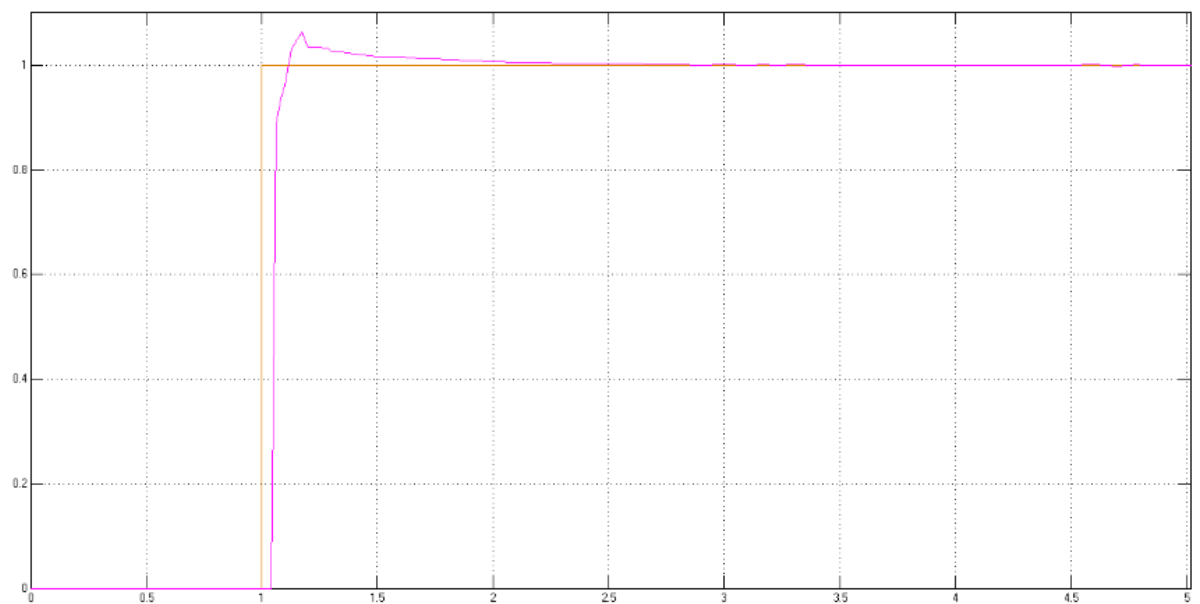


Figura 3.18 Respuesta a lazo cerrado del controlador con planta de primer orden.

Para la Figura 3.18, la respuesta del controlador es menos agresiva, ya que las plantas de primer orden y segundo orden son estables.

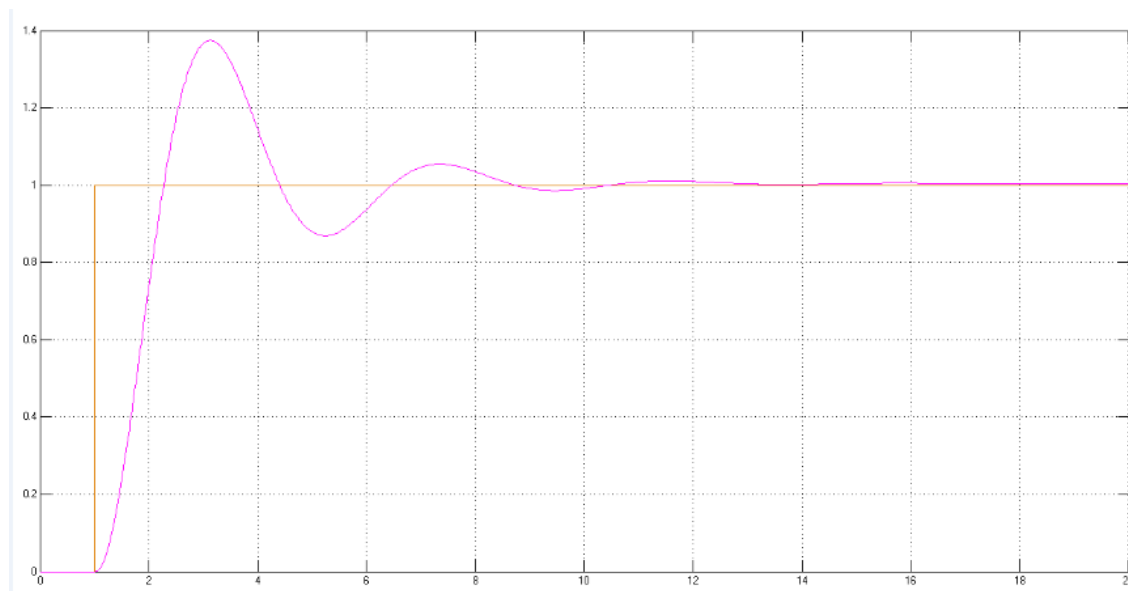


Figura 3.19 Respuesta a lazo cerrado del controlador PID de planta de primer orden integrante.

La forma de onda del controlador para la planta de primer orden integrante (Figura 3.19) es muy similar a la de la Figura 3.17, esta similitud ocurre por la inestabilidad que presentan las plantas con un sistema integrante.

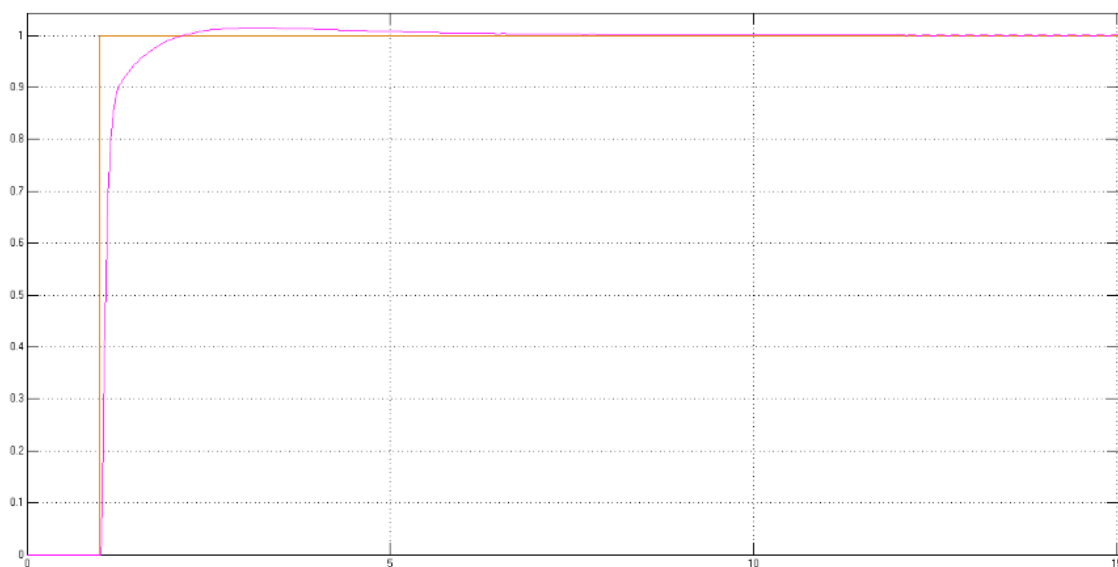


Figura 3.20 Respuesta a lazo cerrado del controlador PID para planta de segundo orden.

Para la Figura 3.20 se muestra que la respuesta del controlador coincide de forma muy parecida a la de la referencia, esto pasa por la estabilidad de estos sistemas. También es notoria la similitud que existe con la respuesta de la Figura 3.18.

3.1.4 Controlador PID en discreto

Ya realizadas las pruebas para los controladores PID en tiempo continuo, ahora se ejecutarán las pruebas para las plantas en discreto con el controlador PID en discreto, para observar y comparar a cada una de ellas.

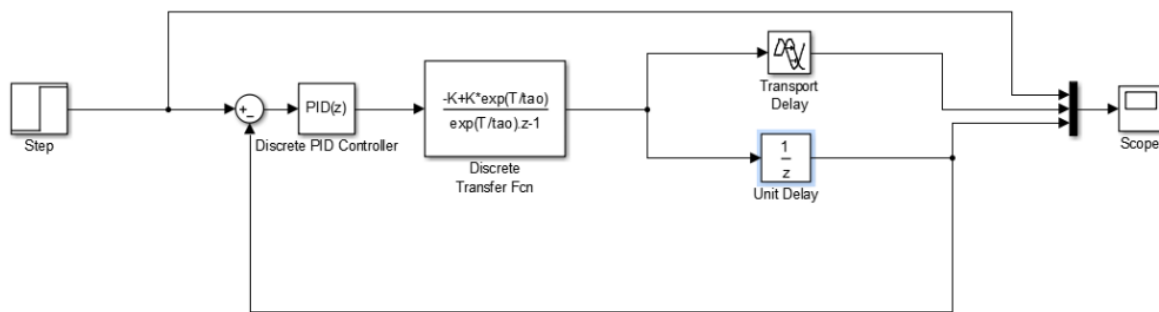


Figura 3.21 Diagrama del controlador PID discreto a lazo cerrado.

La Figura 3.21 representa el diagrama del controlador PID en tiempo discreto, este diagrama como se puede ver es muy parecido al diagrama de la Figura 3.16, las diferencias que se muestran entre ellas es que tanto el controlador como la planta están en tiempo discreto. Este diagrama es el mismo para las todas las plantas.

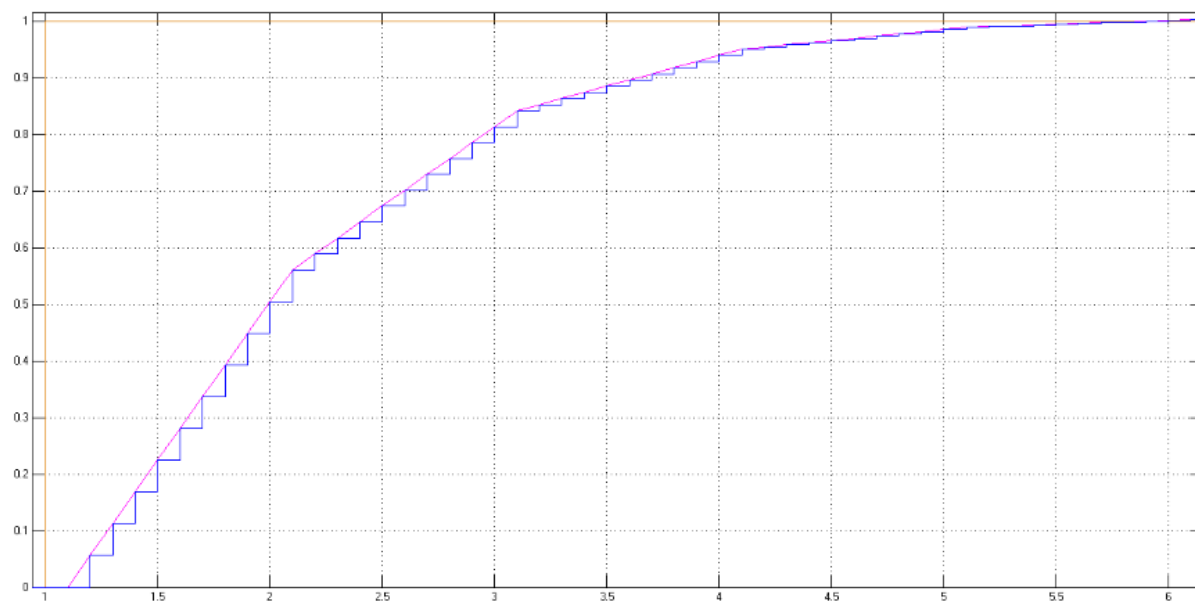


Figura 3.22 Respuesta del controlador PID discreto de la planta integrante.

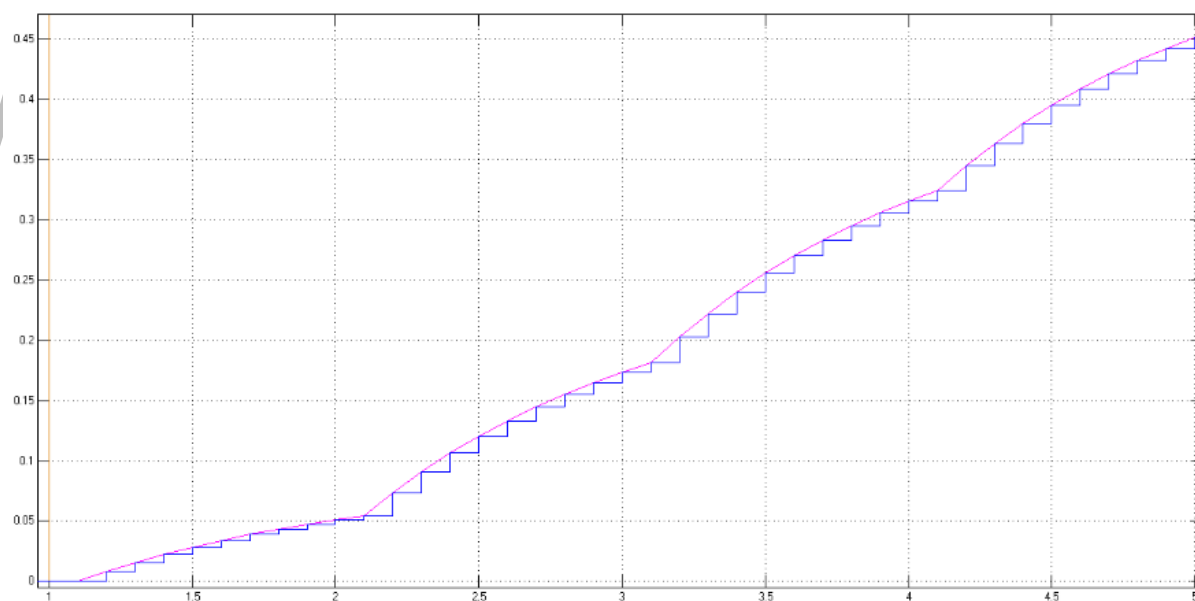


Figura 3.23 Respuesta del controlador PID discreto de la planta de primer orden.

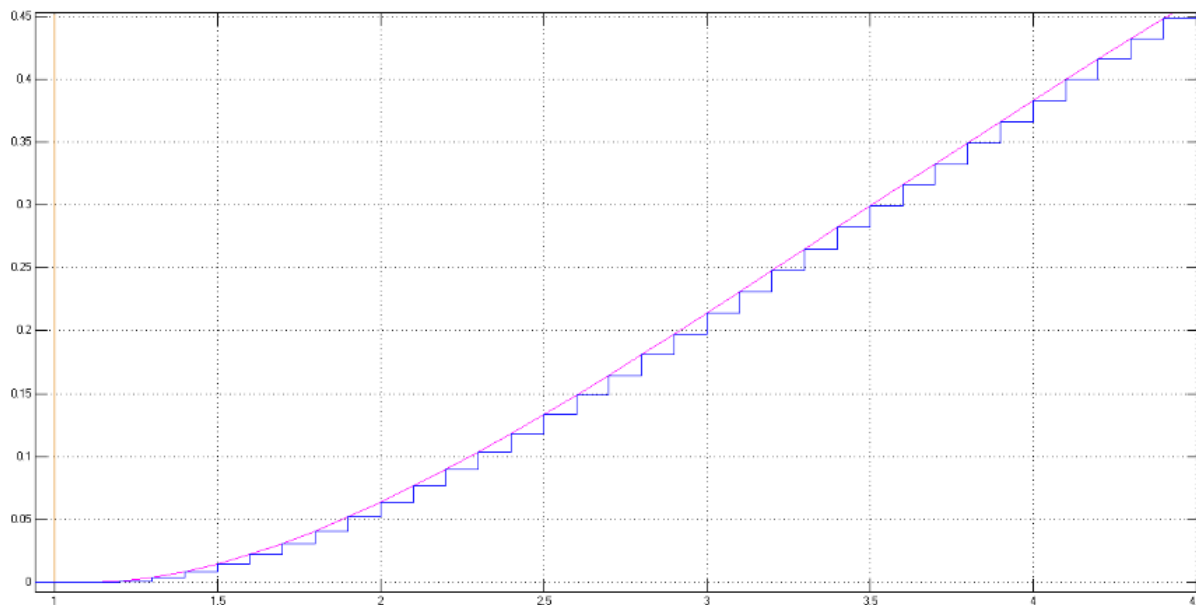


Figura 3.24 Respuesta del controlador PID discreto de la planta de primer orden integrante.

El comportamiento de las plantas con el controlador discreto, es muy similar entre las figuras: Figura 3.22, Figura 3.23 y Figura 3.24, tal y como se observa, la salida del controlador tiende a la de la referencia, mostrando así, control en el sistema.

La sintonización del PID, tiene diferentes parámetros para cada una de ellas, ya que el sistema es distinto y en este caso fueron calculados por una herramienta de Simulink, mientras que los parámetros de la planta, son iguales para las figuras anteriores (Figura 3.22, Figura 3.23 y Figura 3.24).

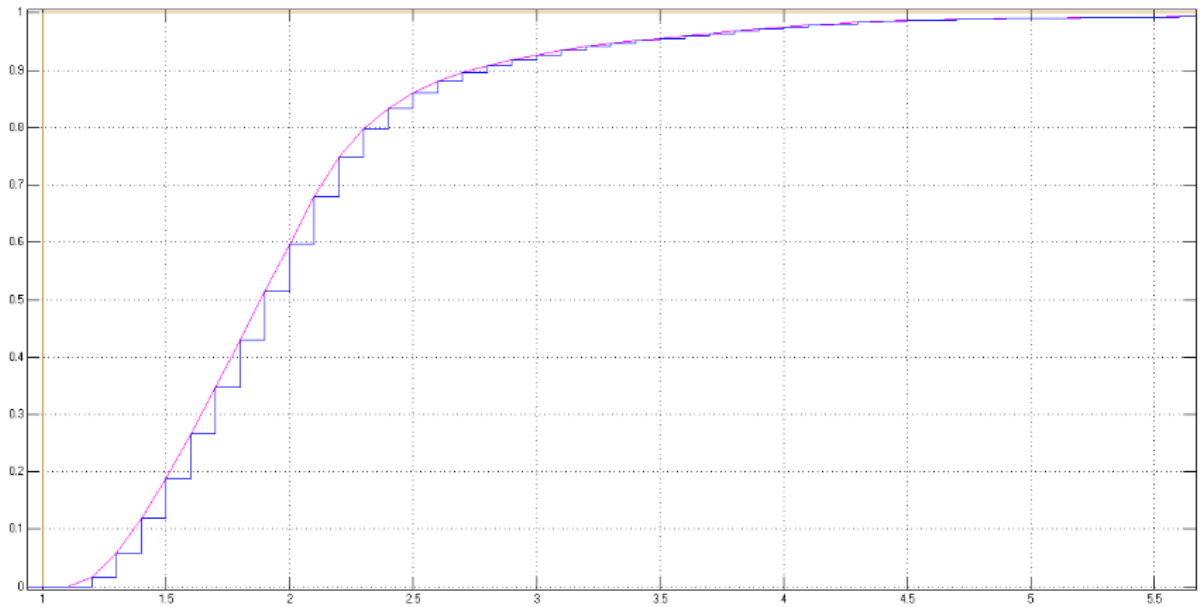


Figura 3.25 Respuesta del controlador PID discreto de la planta de segundo orden.

La Figura 3.25 muestra una reacción de control mucho más rápida que la de las plantas anteriores, ya que su salida se superpone a la referencia a los cinco segundos.

www.bdigital.ula.ve

CAPITULO 4

IMPLEMENTACION DEL CODIGO EN MATLAB Y MICRO

En el presente capítulo se expone el procedimiento para la implementación del código en Matlab y Arduino, así como la comparación entre ellos para verificar la funcionalidad de los sistemas tomando como referencia la respuesta de Matlab.

4.1 IMPLEMENTACION DEL CODIGO EN MATLAB

Para implementar el código de Matlab, se discretizan las ecuaciones de las plantas y luego, se convirtió la función de transferencia discreta en $y(k)$ discreto, para así, obtener los valores últimos, penúltimos y antepenúltimos de los sistemas.

Las funciones de transferencia en tiempo continuo ya están establecidas en las ecuaciones (ecuación (2.1), (2.2) , (2.3) y (2.4)), estas ecuaciones ya discretizadas quedan de la siguiente forma:

$$G(s) = \frac{KT}{z - 1} \quad (4.1)$$

$$G(s) = \frac{K \left(e^{\frac{T}{\tau}} - 1 \right)}{\frac{T}{e^{\tau}z} - 1} \quad (4.2)$$

$$G(s) = \frac{K(z-1) \left(\frac{e^{\frac{T}{\tau}} \tau z}{e^{\frac{T}{\tau}} - 1} + \frac{z(\tau + T - \tau z)}{(z-1)^2} \right)}{z} \quad (4.3)$$

$$G(s) = \frac{Gz + H}{Iz^2 + Jz + L} \quad (4.4)$$

La ecuación (4.1) representa la forma de la planta integrante en tiempo discreto, la planta de primer orden discretizada se muestra en la ecuación (4.2), la planta de primer orden integrante en la ecuación (4.3) y la ecuación (4.4) sería la planta de segundo orden en discreto.

La manera más simple que se utilizó para procesar estas ecuaciones a través de un código en Matlab, consistió en transformar dichas ecuaciones a $y(k)$ en discreto, para utilizar los valores pasados de cada uno de ellos.

Para la ecuación (4.5) el valor de $a = K * T$, mientras que $(k - 1)$ es el valor último, por lo cual es la ecuación que se presenta al momento de realizar el código en Matlab. Para ver la respuesta de la curva y compararla con Simulink Matlab, se establecen valores para así cotejar la validación del código. Los valores utilizados para la ganancia $K = 1$ y el tiempo de muestreo $T = 0.1$, obteniendo así la Figura 4.1.

$$y(k) = y(k - 1) + a * u(k - 1) \quad (4.5)$$

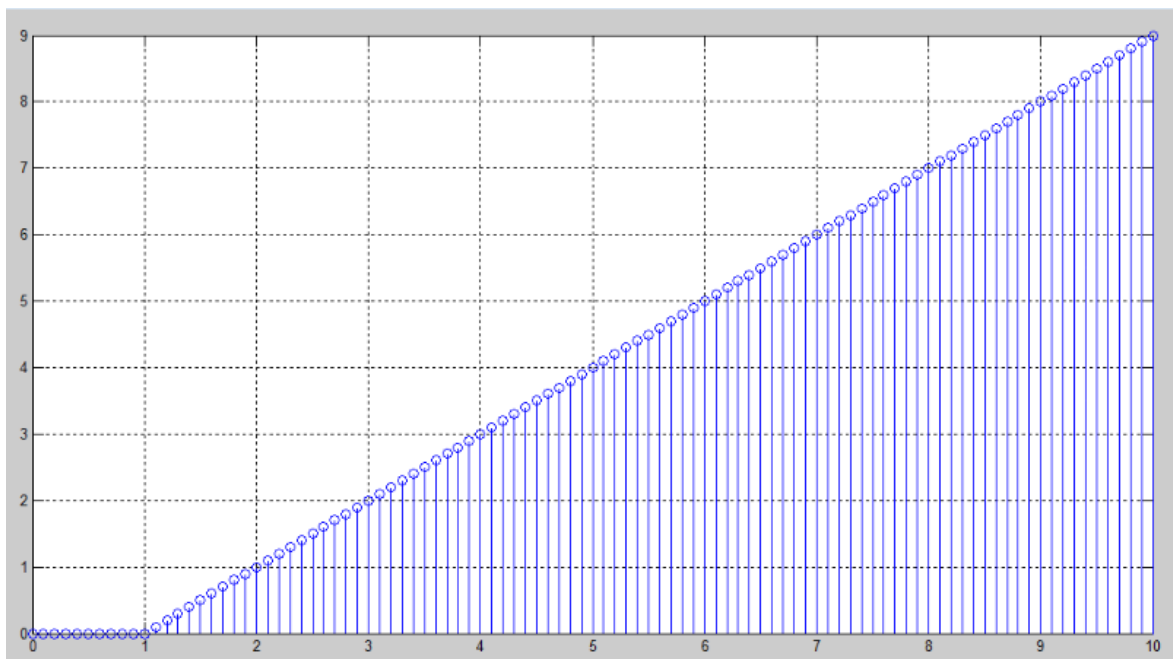


Figura 4.1 Respuesta de la planta integrante en código de Matlab.

El valor de $a = K \left(e^{\frac{T}{\tau}} - 1 \right)$ y $b = e^{\frac{T}{\tau}}$ para la ecuación (4.6) y la gráfica correspondiente para esta ecuación es la Figura 4.2, la representación de la gráfica como de la ecuación pertenece a la planta de primer orden en $y(k)$ discreto, el valor de ganancia que se usa es de $K=1$, tiempo de muestreo $T = 0.1$ y $\tau = 1$.

$$y(k) = \frac{y(k-1) + a * u(k-1)}{b} \quad (4.6)$$

Como se muestra en la Figura 4.2 y la Figura 3.4, son prácticamente las mismas formas de ondas, esta sería la garantía de la efectividad que posee el código realizado en Matlab.

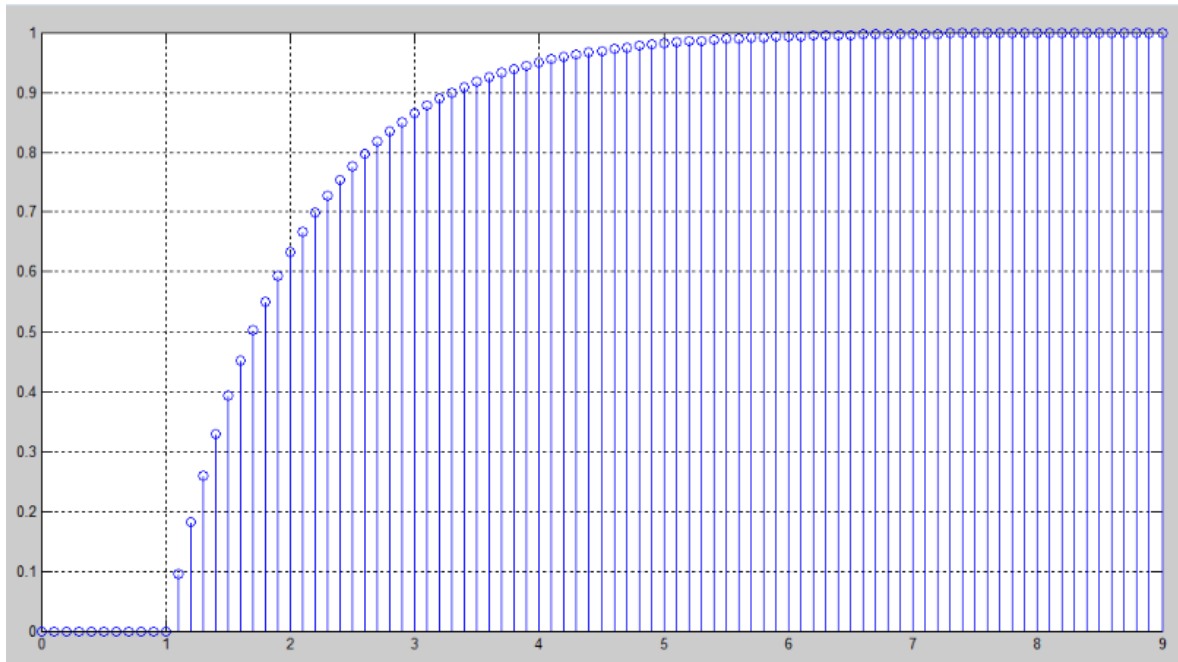


Figura 4.2 Respuesta de la planta de primer orden en código de Matlab.

La ecuación (4.7) pertenece a la planta de primer orden integrante en $y(k)$ en tiempo discreto, esta ecuación es demasiado amplia, por lo que se realizó un cambio de variable, para simplificar la ecuación. El valor de $a = K \left(e^{\frac{T}{\tau}T} - e^{\frac{T}{\tau}\tau} + \tau \right)$, $b = K \left(e^{\frac{T}{\tau}\tau} - \tau - T \right)$, $c = e^{\frac{T}{\tau}}$ y $d = e^{\frac{T}{\tau}} + 1$, los valores de los parámetros de la planta que se asignaron para poder visualizar la forma de onda son: la ganancia $K = 1$, el tiempo de muestreo $T = 0.5$ y $\tau = 1$.

$$y(k) = \frac{a * u(k - 1) + b * u(k - 2) + d * y(k - 1) - y(k - 2)}{c} \quad (4.7)$$

La Figura 3.6 representa la curva de la planta de primer orden integrante al igual que la Figura 4.3, al comparar ambas figuras se nota que similitud que hay entre ellas, lo que se podría decir que son las mismas curvas, a pesar de que se utilizan diferentes herramientas y deja como validación lo veracidad del código.

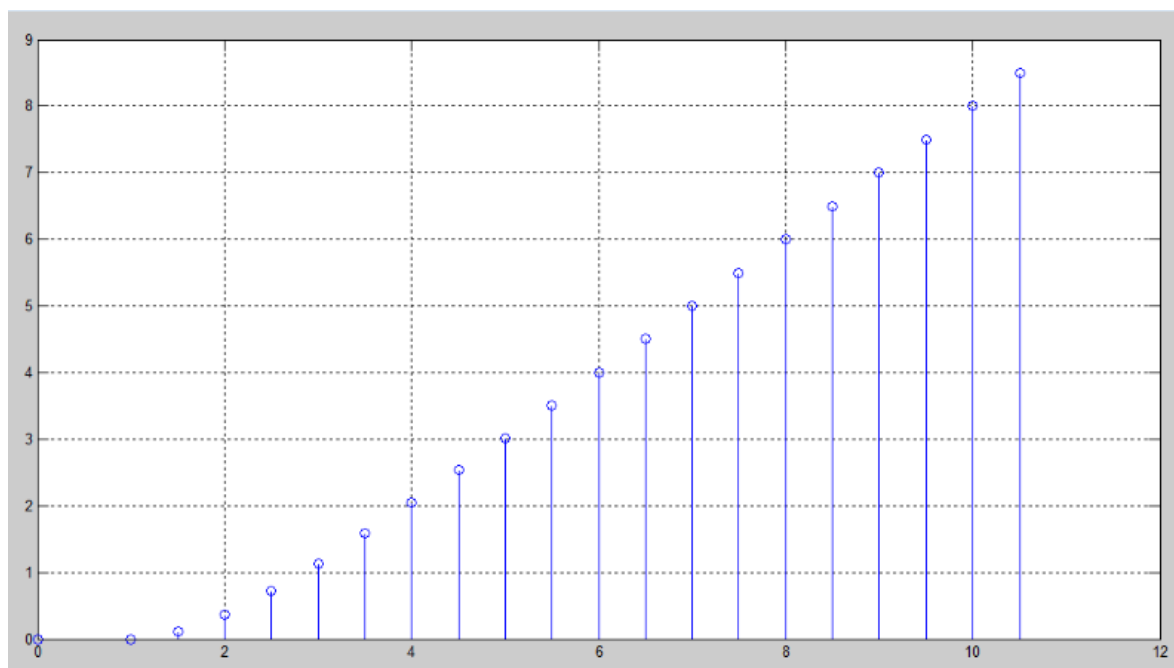


Figura 4.3 Gráfica de la planta de primer orden integrante en código de Matlab.

$$y(k) = \frac{G * u(k-1) + H * u(k-2) - J * y(k-1) - L * y(k-2)}{I} \quad (4.8)$$

La ecuación (4.8) representa la planta de segundo orden, esta se aplicó directamente al código de Arduino.

A continuación, las gráficas que presentan las curvas de respuesta que se obtienen del código realizado en Matlab, aplicando el mismo código anterior, pero esta vez, se agrega el tiempo muerto o de retardo.

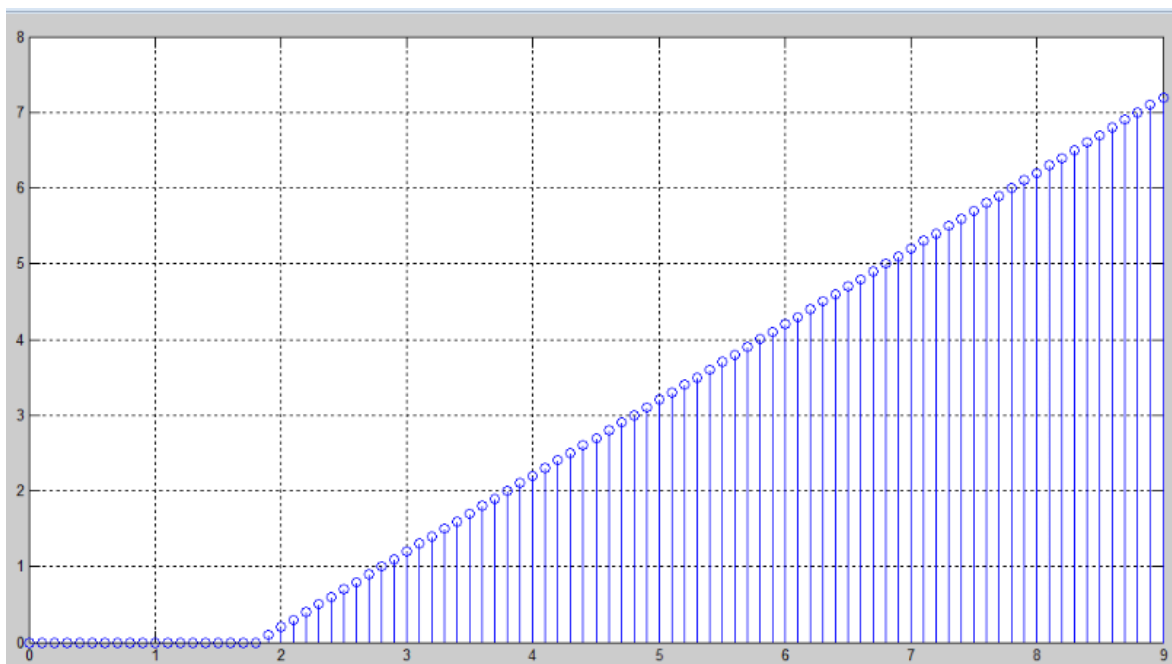


Figura 4.4 Respuesta de la planta integrante con tiempo muerto.

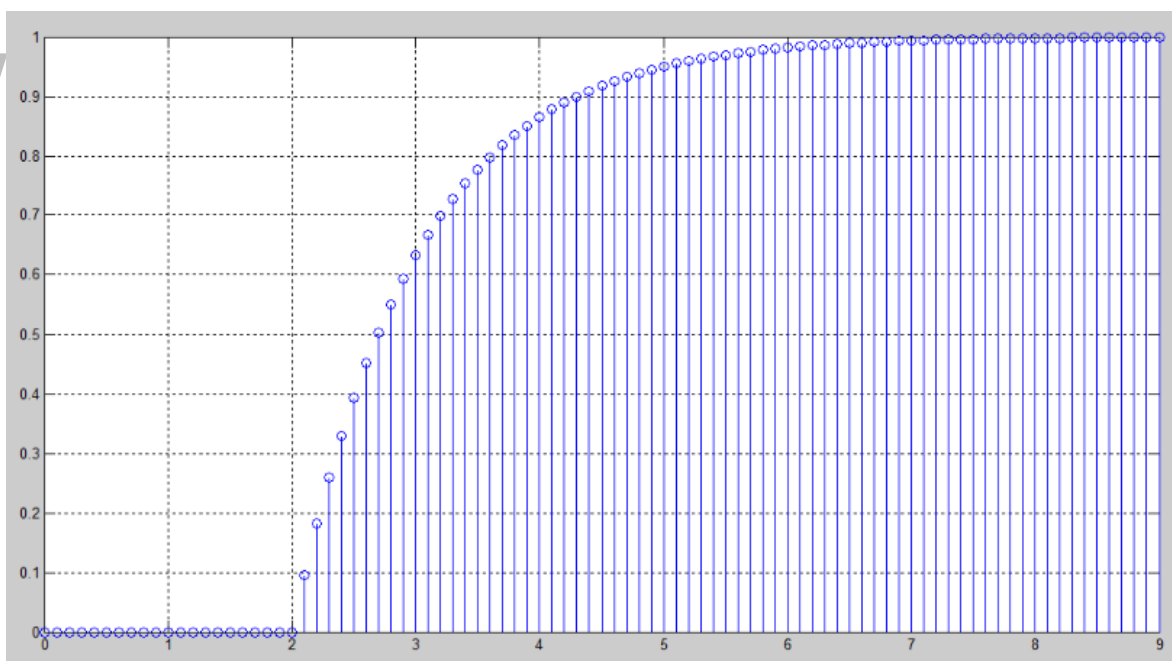


Figura 4.5 Respuesta de la planta de primer orden con tiempo muerto.

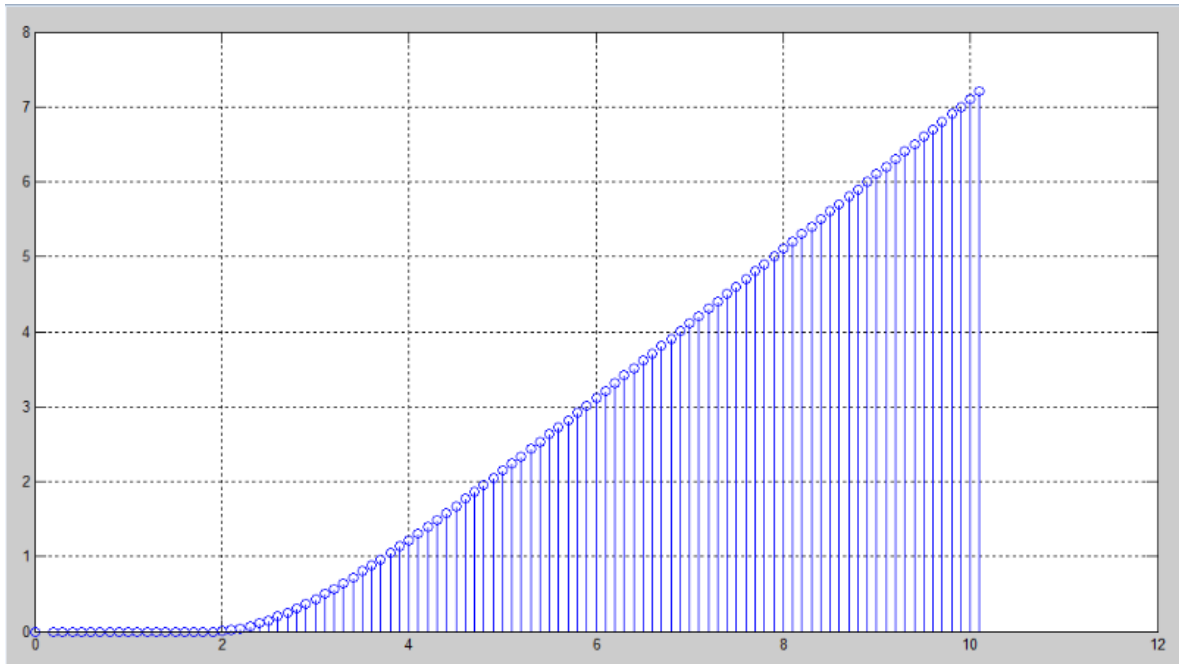


Figura 4.6 Respuesta de la planta de primer orden integradora con tiempo muerto.

Al comparar la Figura 4.4, Figura 4.5 y Figura 4.6 con las figuras (Figura 4.1, Figura 4.2 y Figura 4.3), la forma de onda que se adquieren para cada una de las plantas y que al compararlas entre ellas mismas, tienen la misma curva de respuesta, con la diferencia que están desplazadas un segundo, esto se debe a que el valor de tiempo muerto o de retardo aplicado es de $t_0 = 1$, por consiguiente, ambos códigos están bien planteados.

El código final que se implementó en Matlab para adecuar el tiempo muerto por desplazamiento con líneas de retardo, se realizó por medio de vectores con una función de desplazamiento, el cual desplaza el vector a medida que se mueve en el tiempo, este procedimiento se llevó a cabo, debido a que se desechó el tiempo muerto por aproximación. Al realizarse las pruebas, se verifica que el código planteado funciona correctamente, ya que arrojó las mismas graficas anteriores y este código es adaptable para ser aplicado en el micro ESP32.

4.2 MICROCONTROLADOR ESP32

Es una serie de microcontroladores de bajo costo y de bajo consumo con sistema en chip el cual posee WI-FI y Bluetooth de modo dual integrados, creado y desarrollado por *Espressif*

Systems. Emplea dos microprocesadores de bajo consumo Tensilica Xtensa de 32 Bits LX6 en sus variantes de simple y doble núcleo de procesamiento cuyas frecuencias pueden controlarse independientemente entre 80 megahercios (MHz) y 240 MHz.

4.2.1 Características del micro ESP32

- Procesador:
 - CPU: microprocesador de 32-bit Xtensa LX6 de doble núcleo (o de un solo núcleo), operando a 160°MHz ó 240 MHz y rindiendo hasta 600 DMIPS.
 - Co-procesador de ultra baja energía (ULP).
- Memoria: 520 KiB SRAM
- Conectividad inalámbrica:
 - Wi-Fi: 802.11 b/g/n
 - Bluetooth: v4.2 BR/EDR y BLE

4.2.2 Interfaces periféricas

Incluye seguridad criptográfica por *hardware*, un coprocesador para el modo bajo consumo e interfaces periféricas como: I2C, SPI, Can; PWM, entre otros.

4.3 IMPLEMENTACION DEL CÓDIGO PARA EL MICRO ESP32

La implementación del código en Arduino se mostrará a continuación a través de diagramas de flujo, estos explicaran la ejecución del mismo.

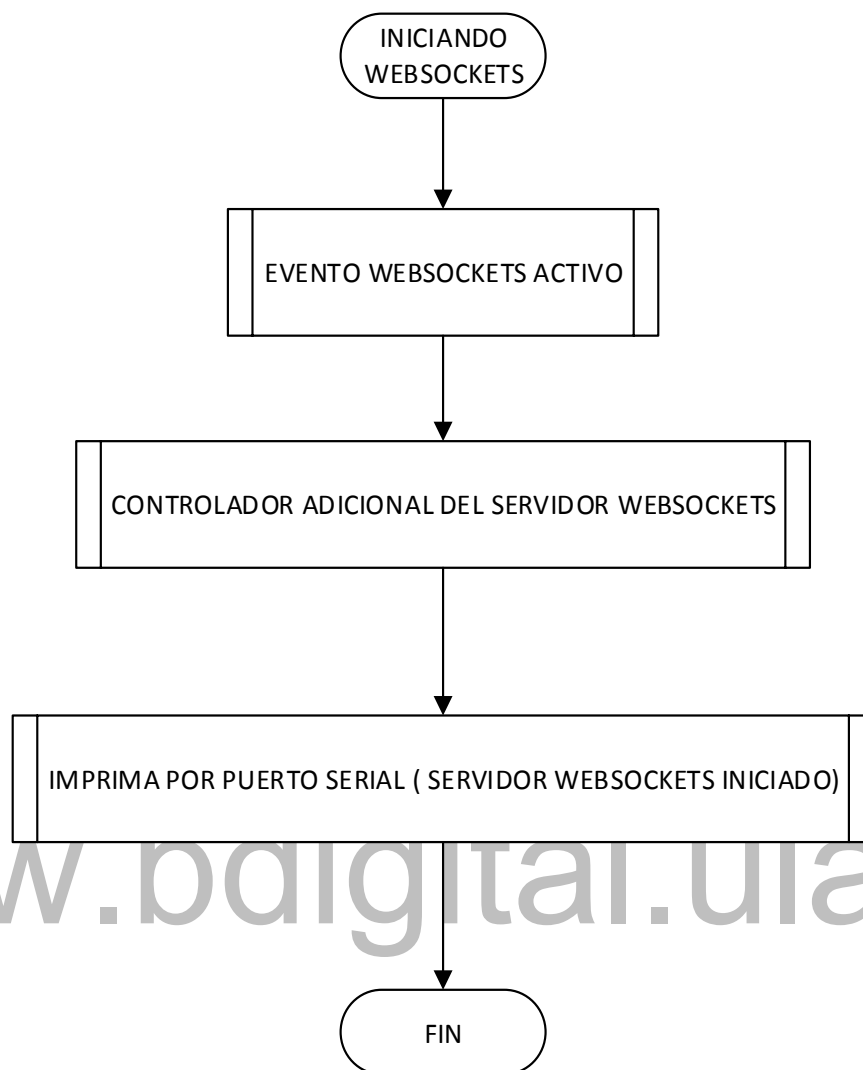


Figura 4.7 Inicialización de *WebSockets*

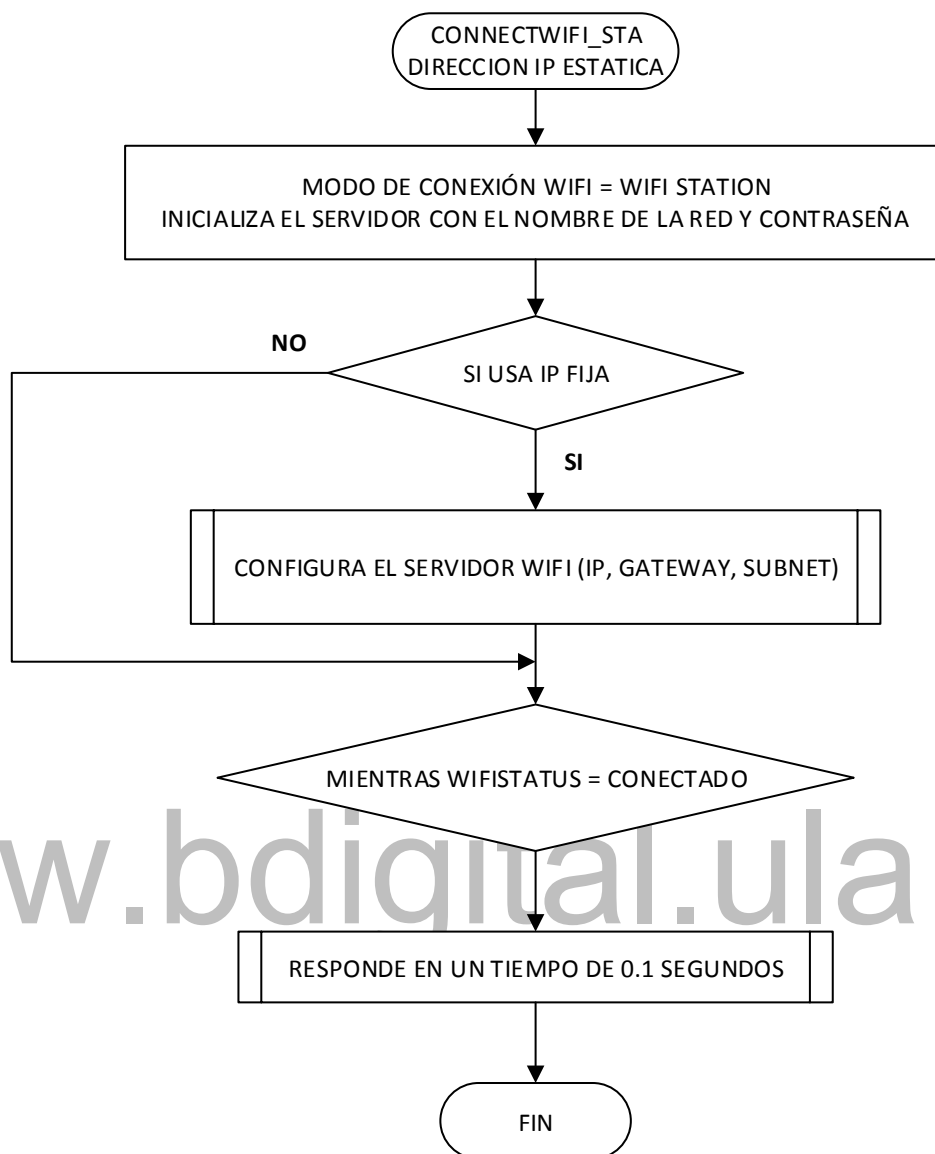


Figura 4.8 Conexión al router.

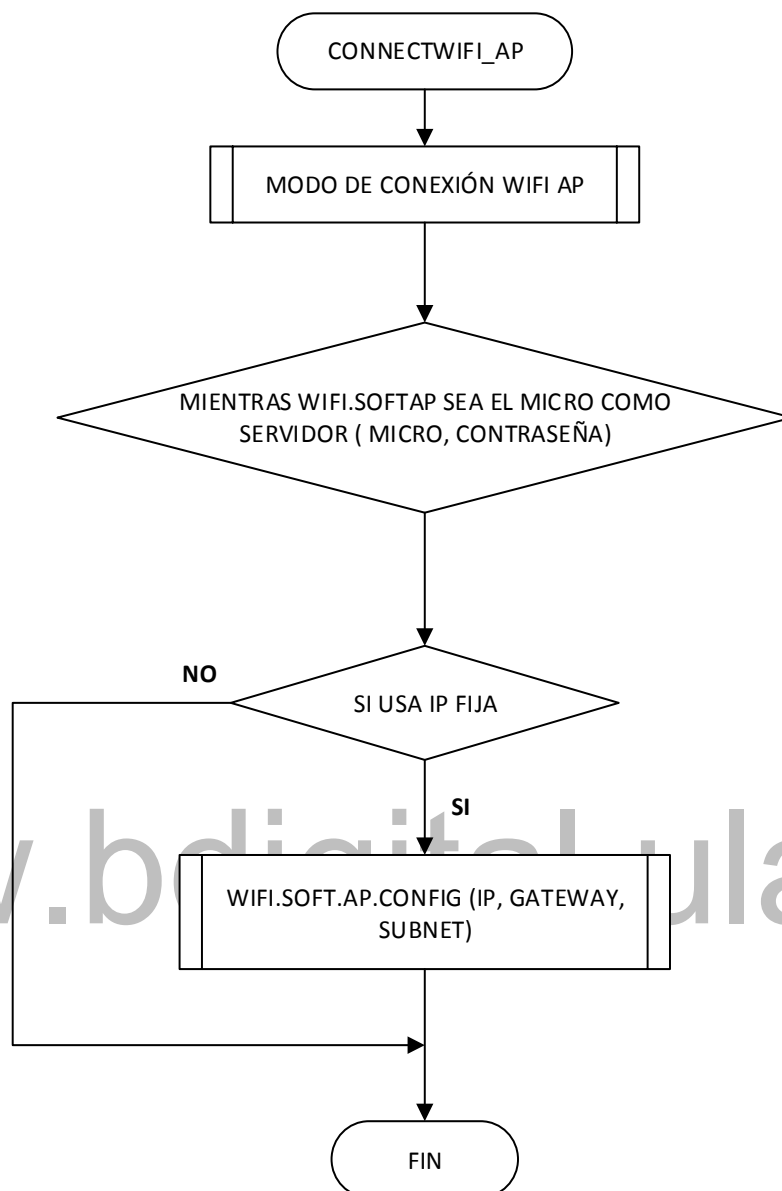


Figura 4.9 Conexión como punto de acceso al módulo ESP32.

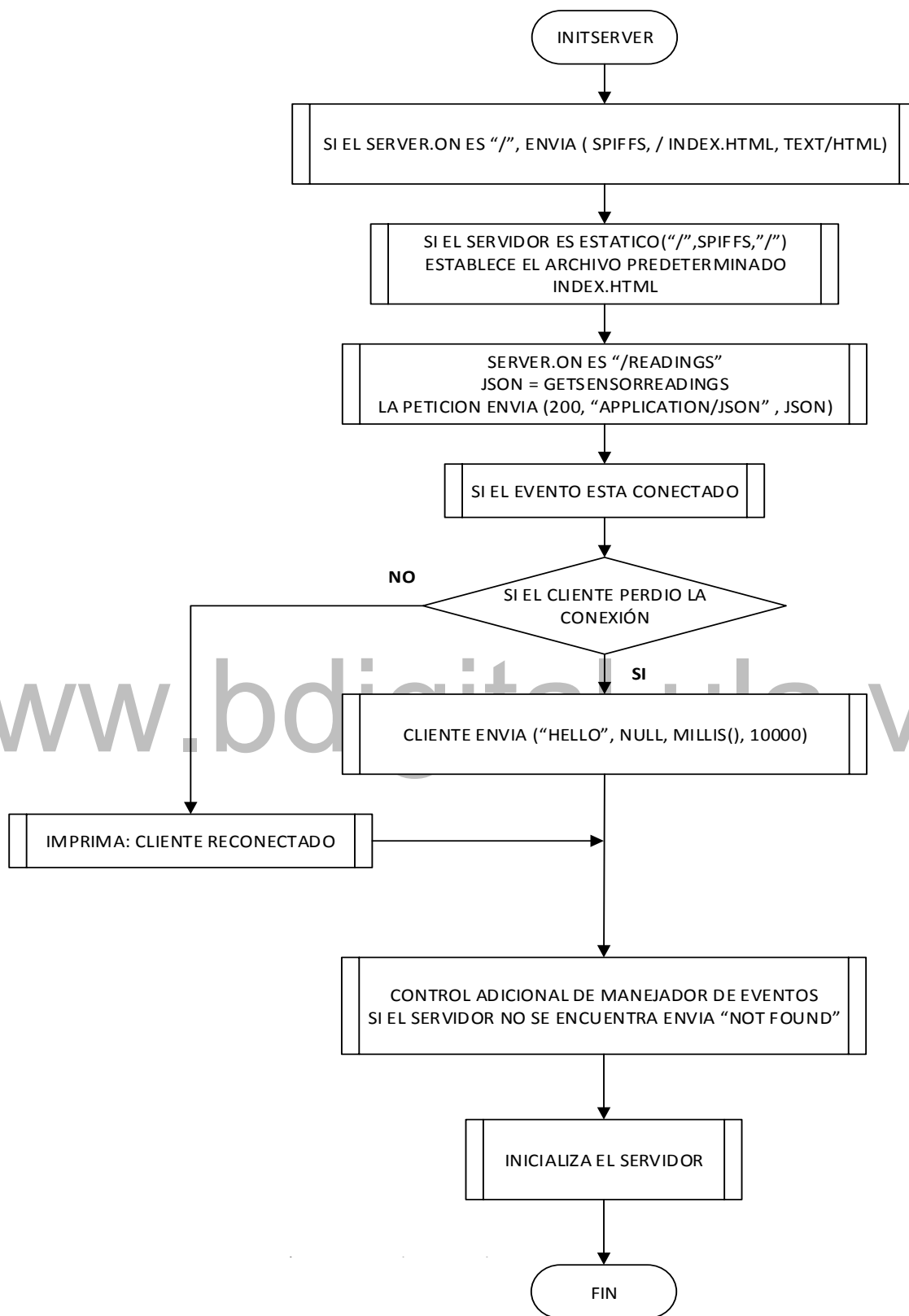


Figura 4.10 Inicialización del servidor.

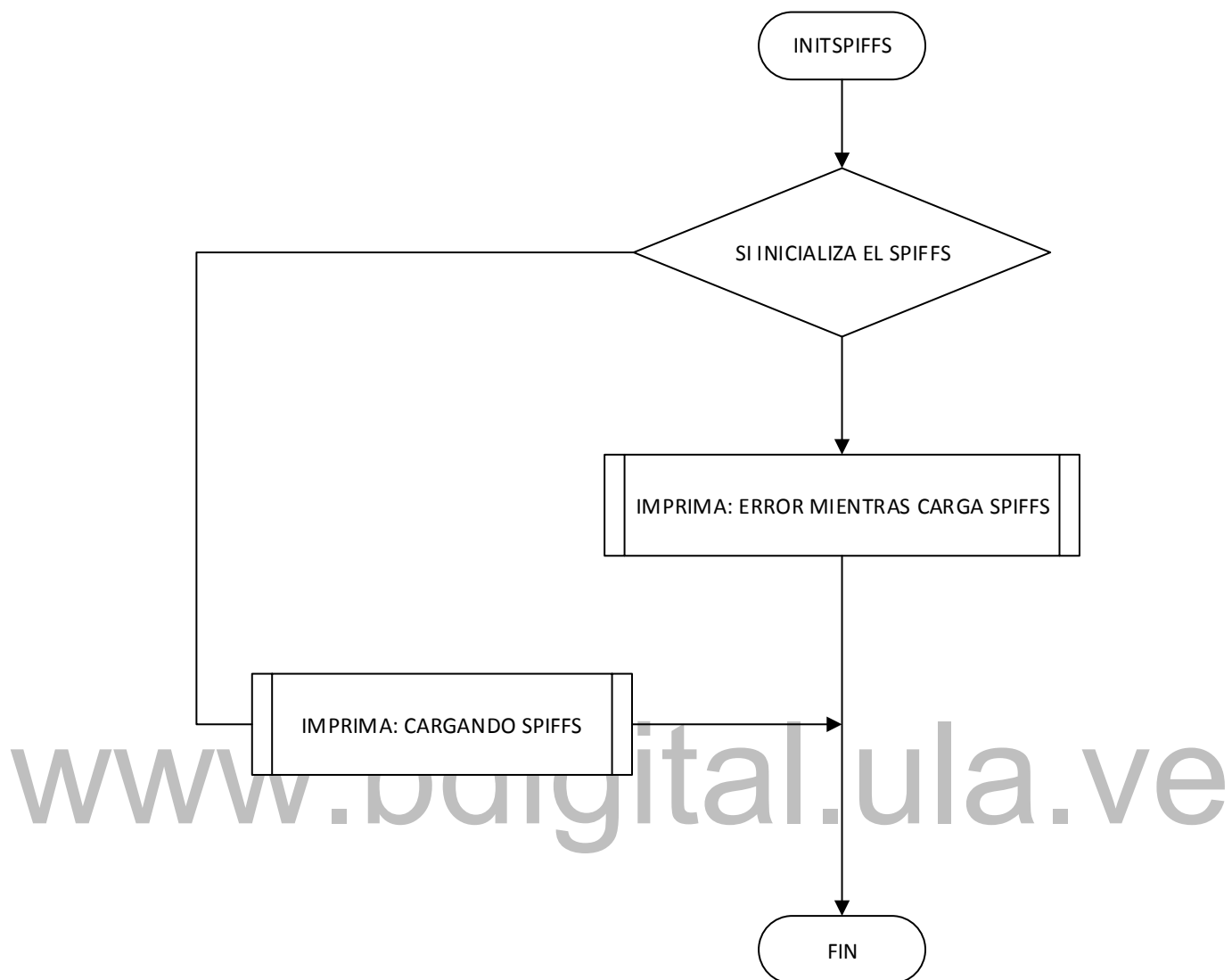


Figura 4.11 Inicialización del *SPIFFS*.

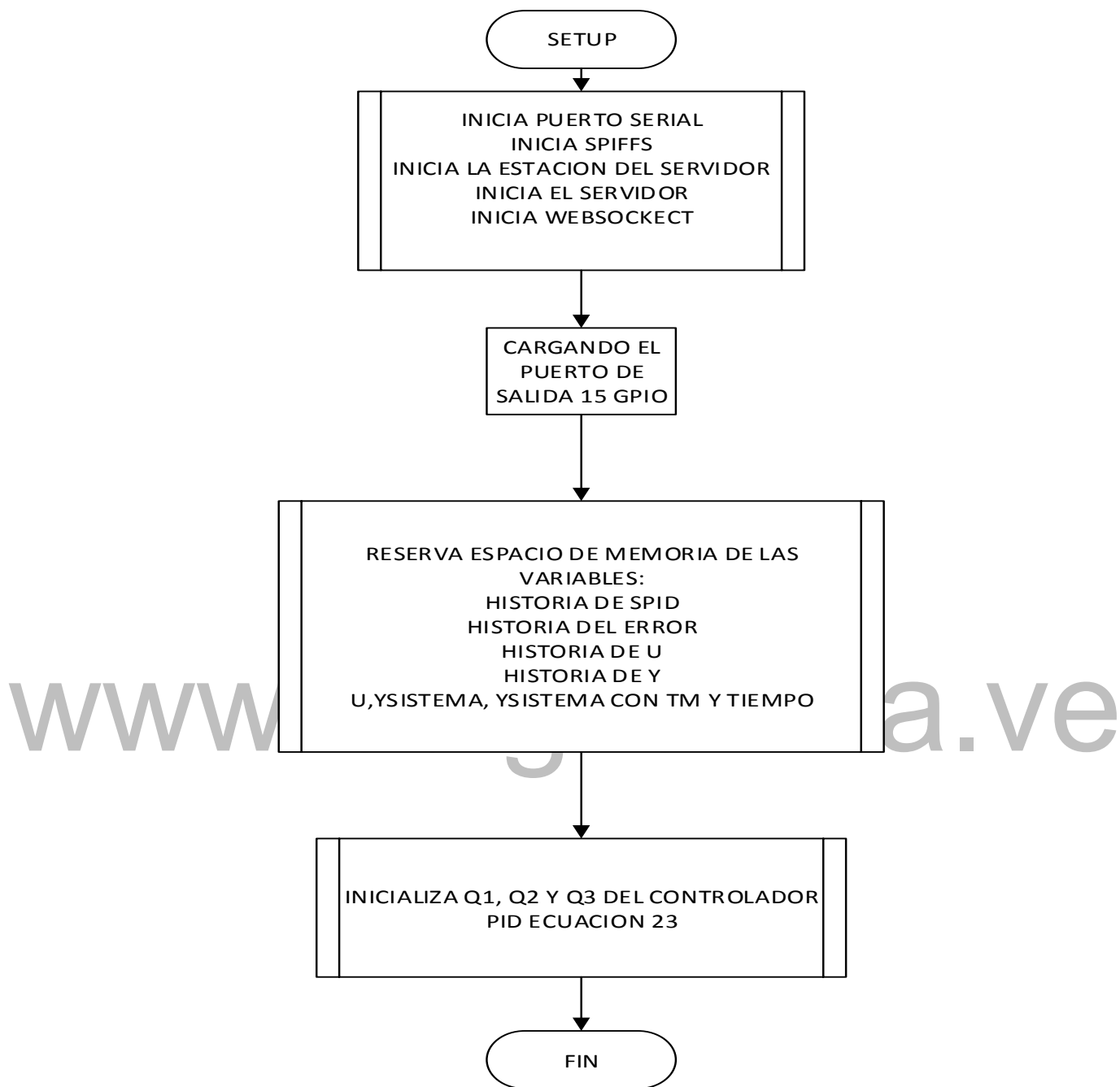


Figura 4.12 Función *setup*.

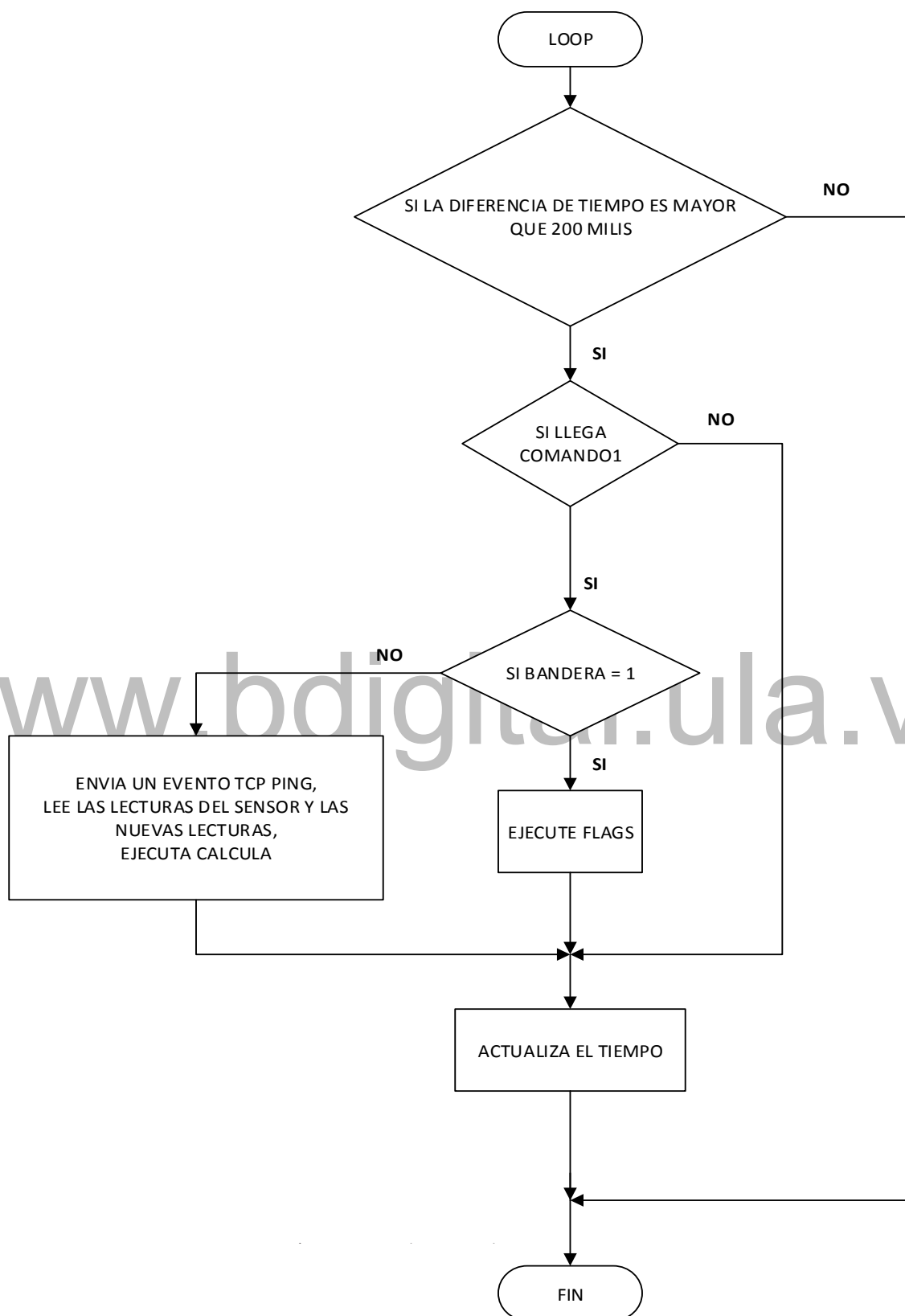


Figura 4.13 Función *loop*.

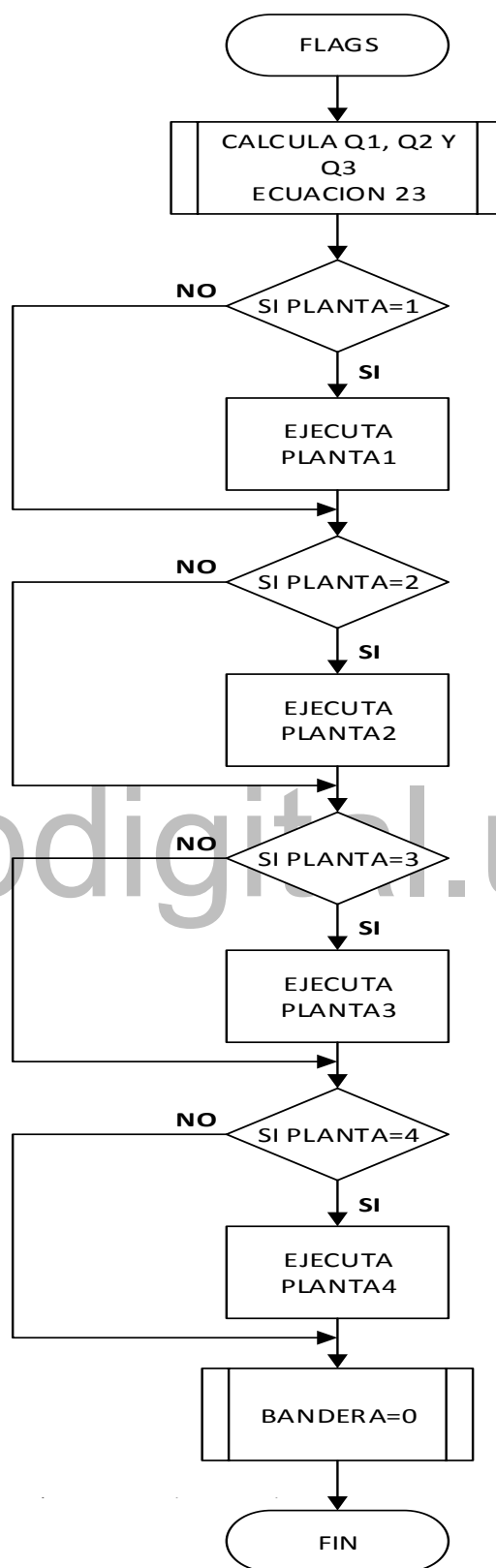


Figura 4.14 Función *flags* para ejecutar tipo de planta.

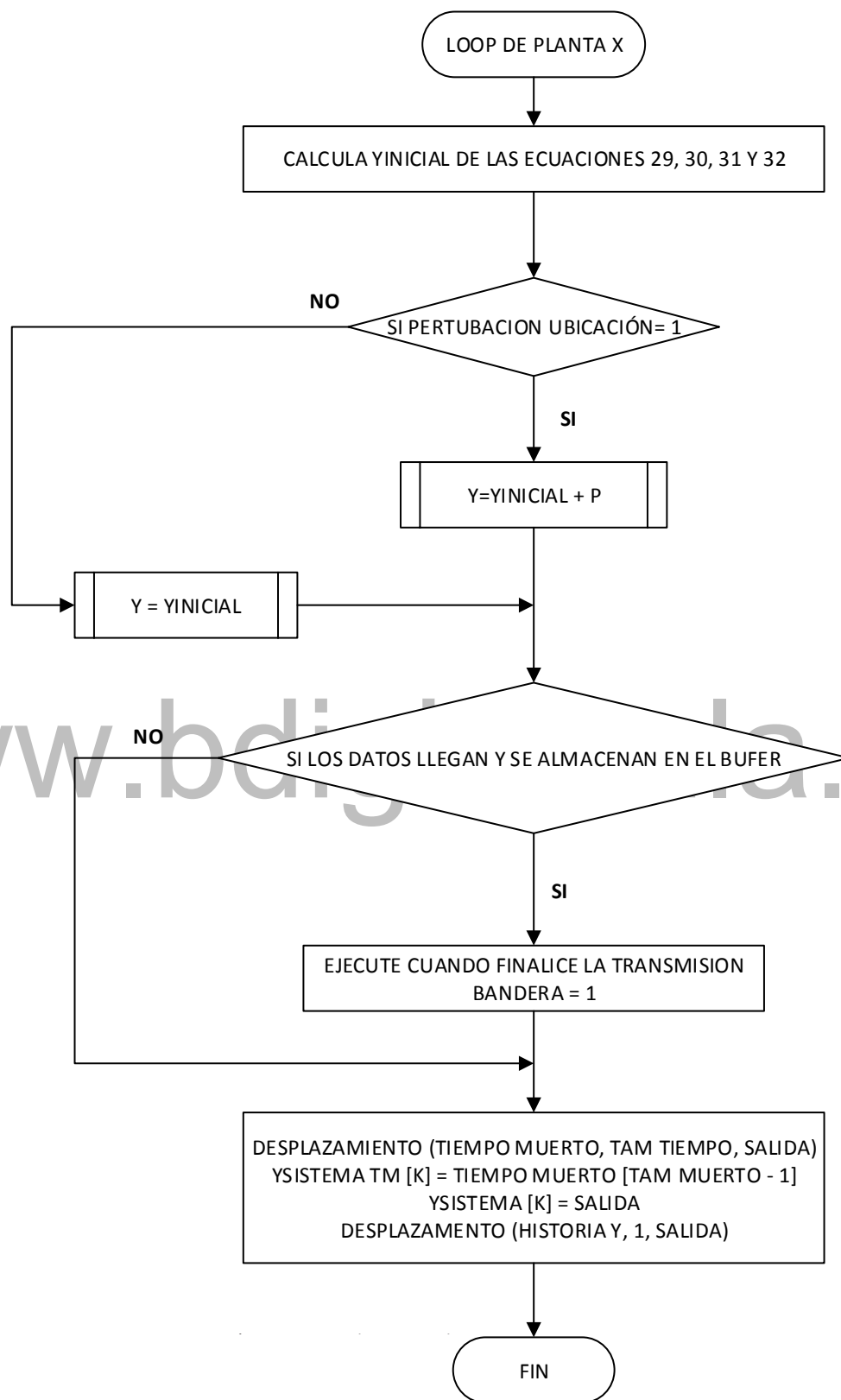


Figura 4.15 Cálculo de la salida de la planta con o sin perturbaciones.

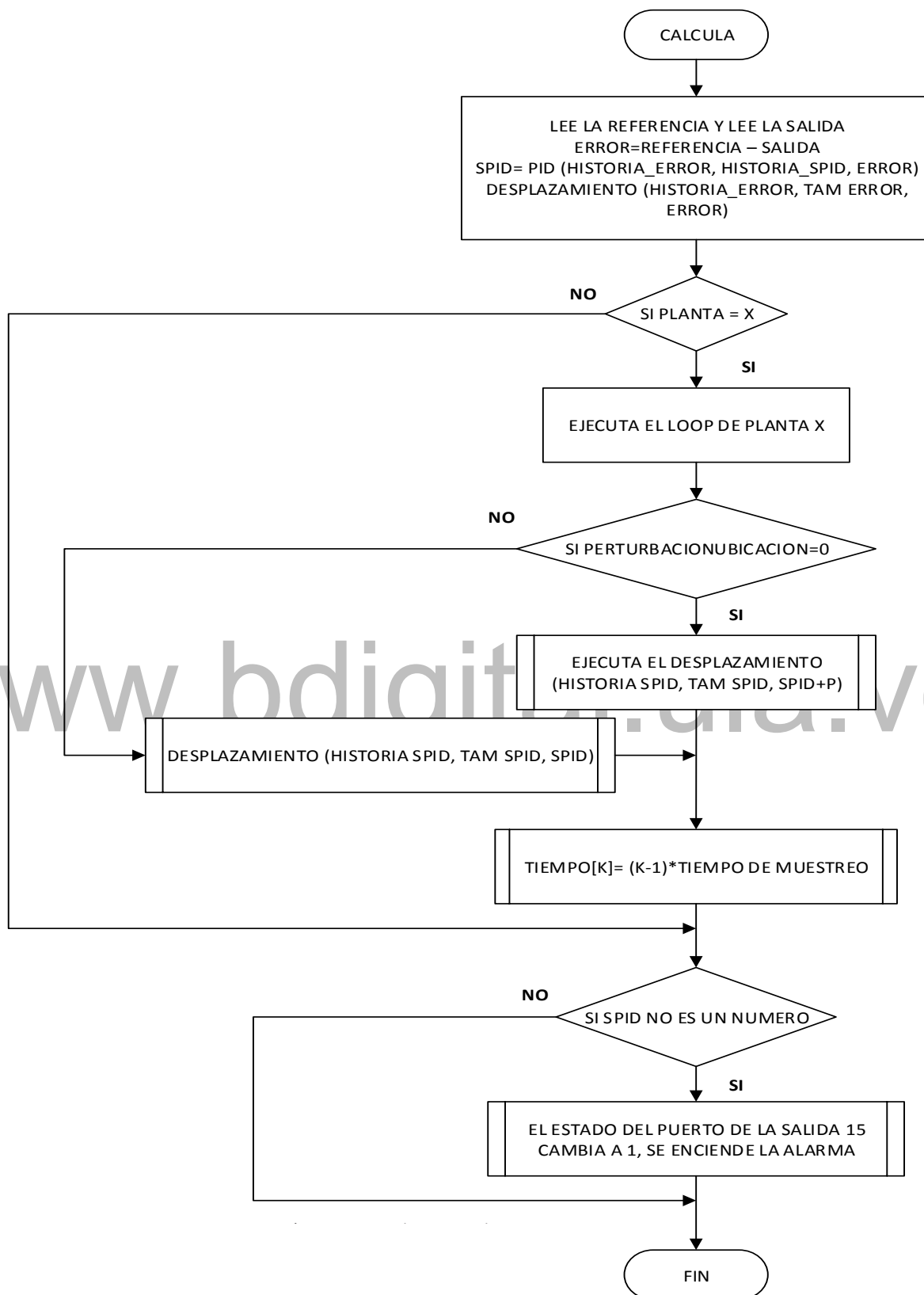


Figura 4.16 Función de cálculos de las plantas y desplazamiento.

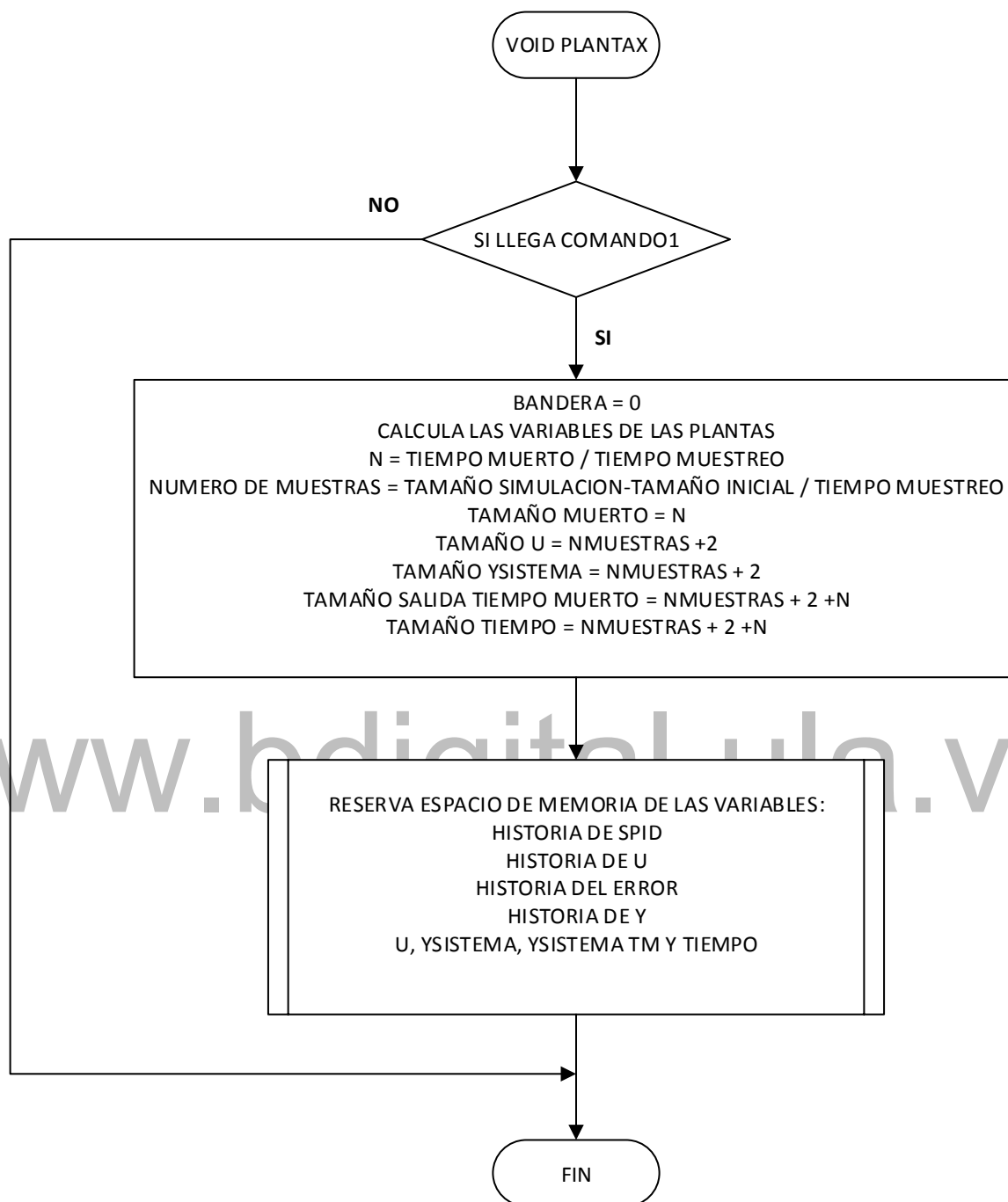


Figura 4.17 Cálculo de variables de las plantas y reserva de memoria.

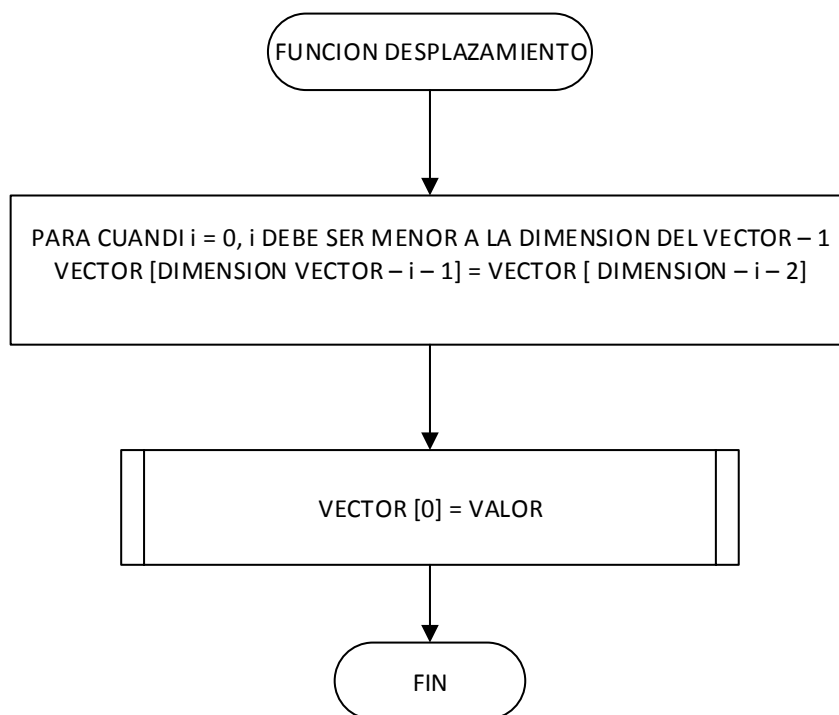


Figura 4.18 Función de desplazamiento por vectores.

www.bdigital.ula.ve

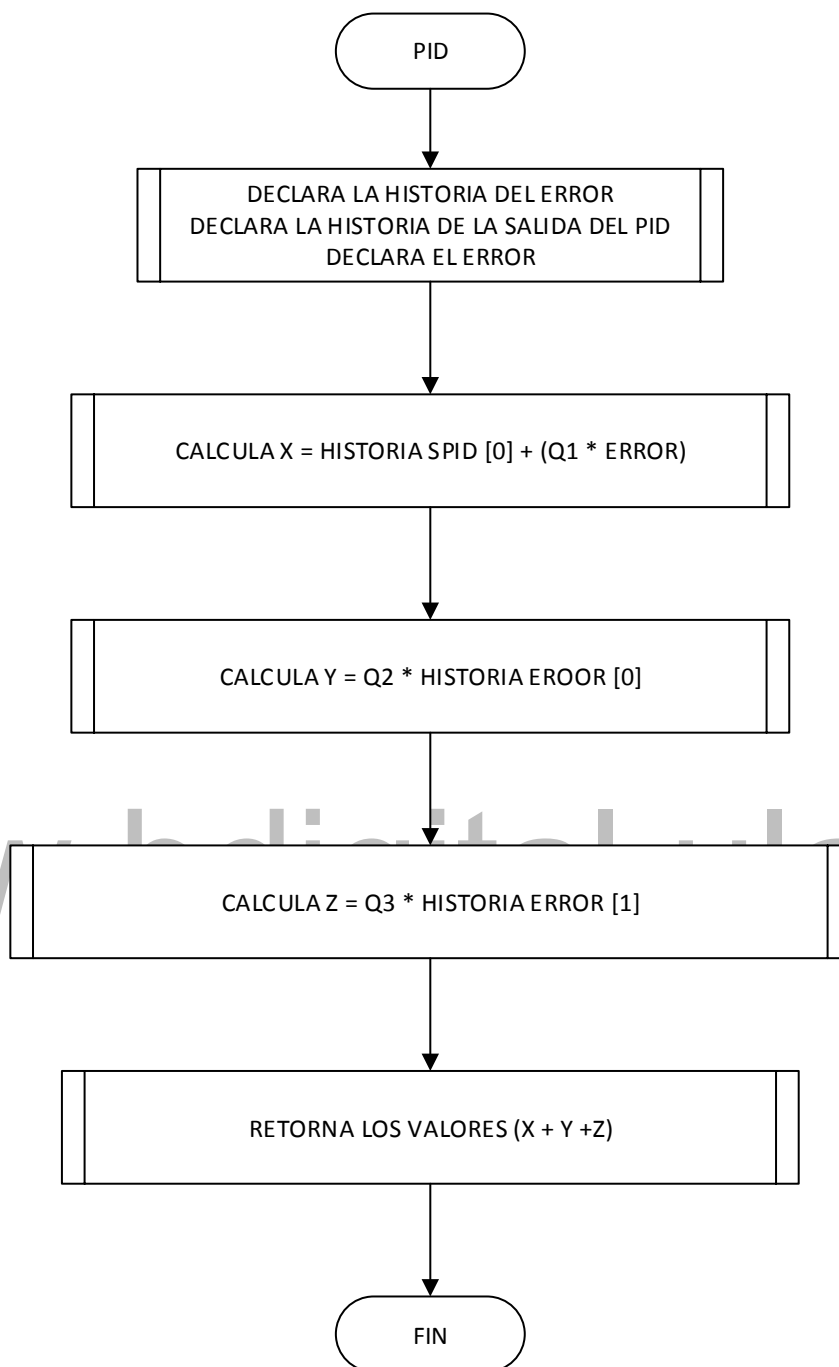


Figura 4.19 Cálculo del controlador PID.

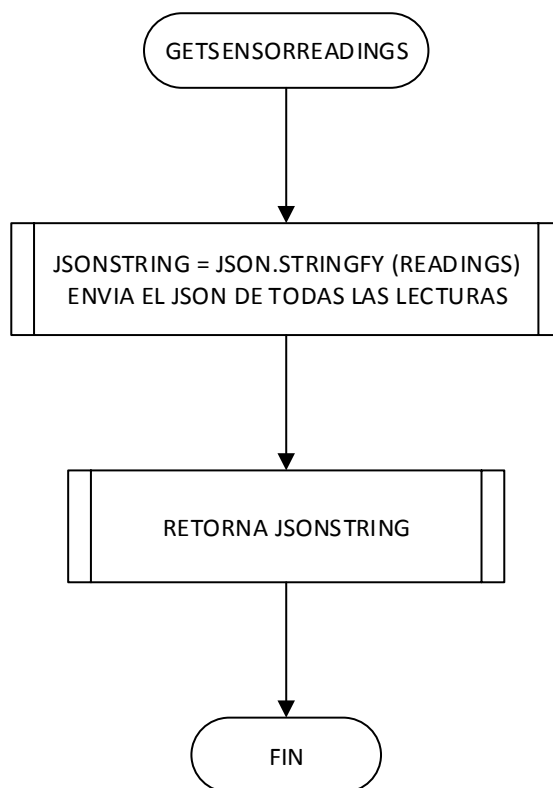


Figura 4.20 Envío de lecturas para la gráfica.

www.bdigital.ula.ve

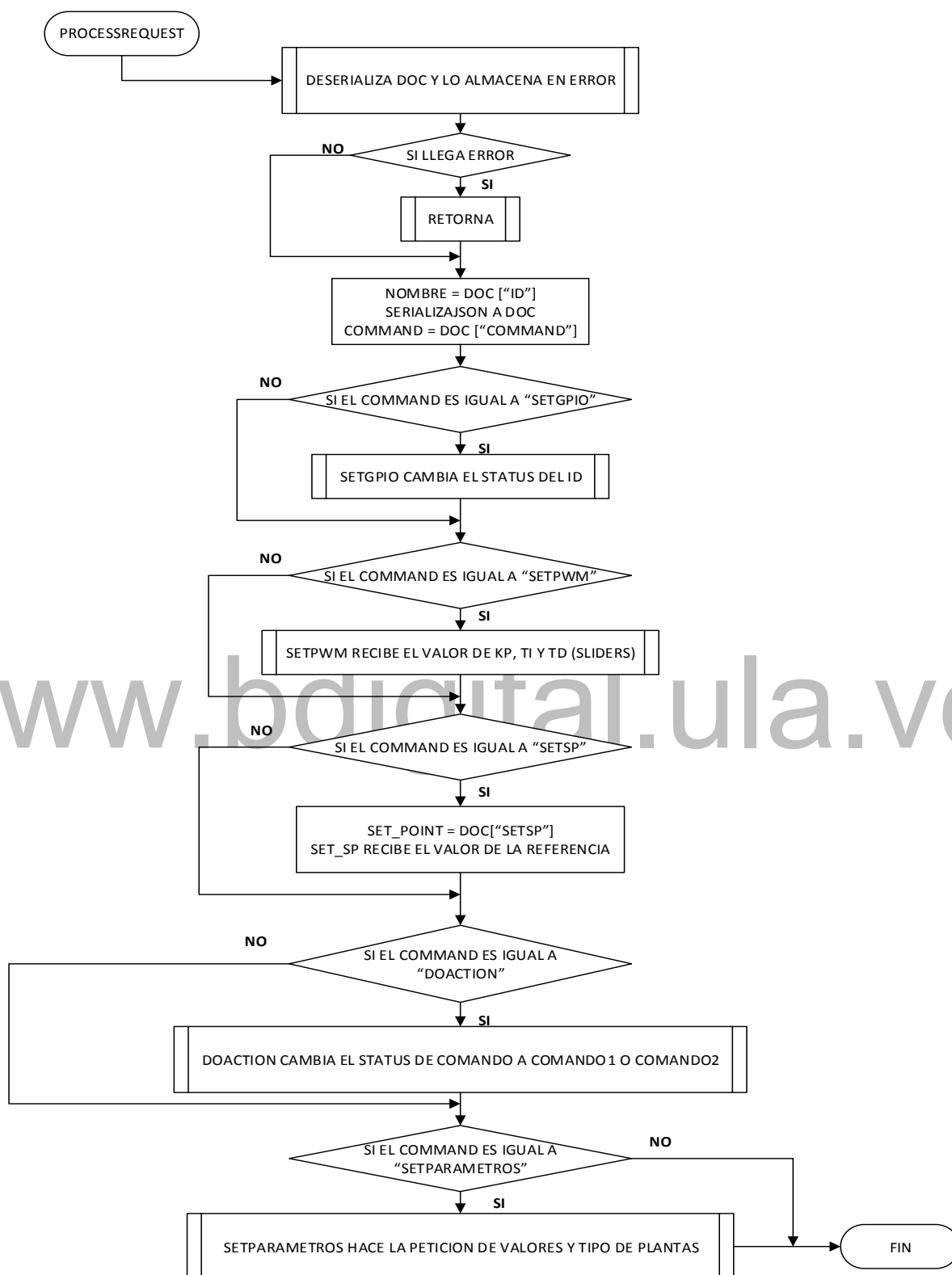


Figura 4.21 Petición de parámetros de la interfaz.

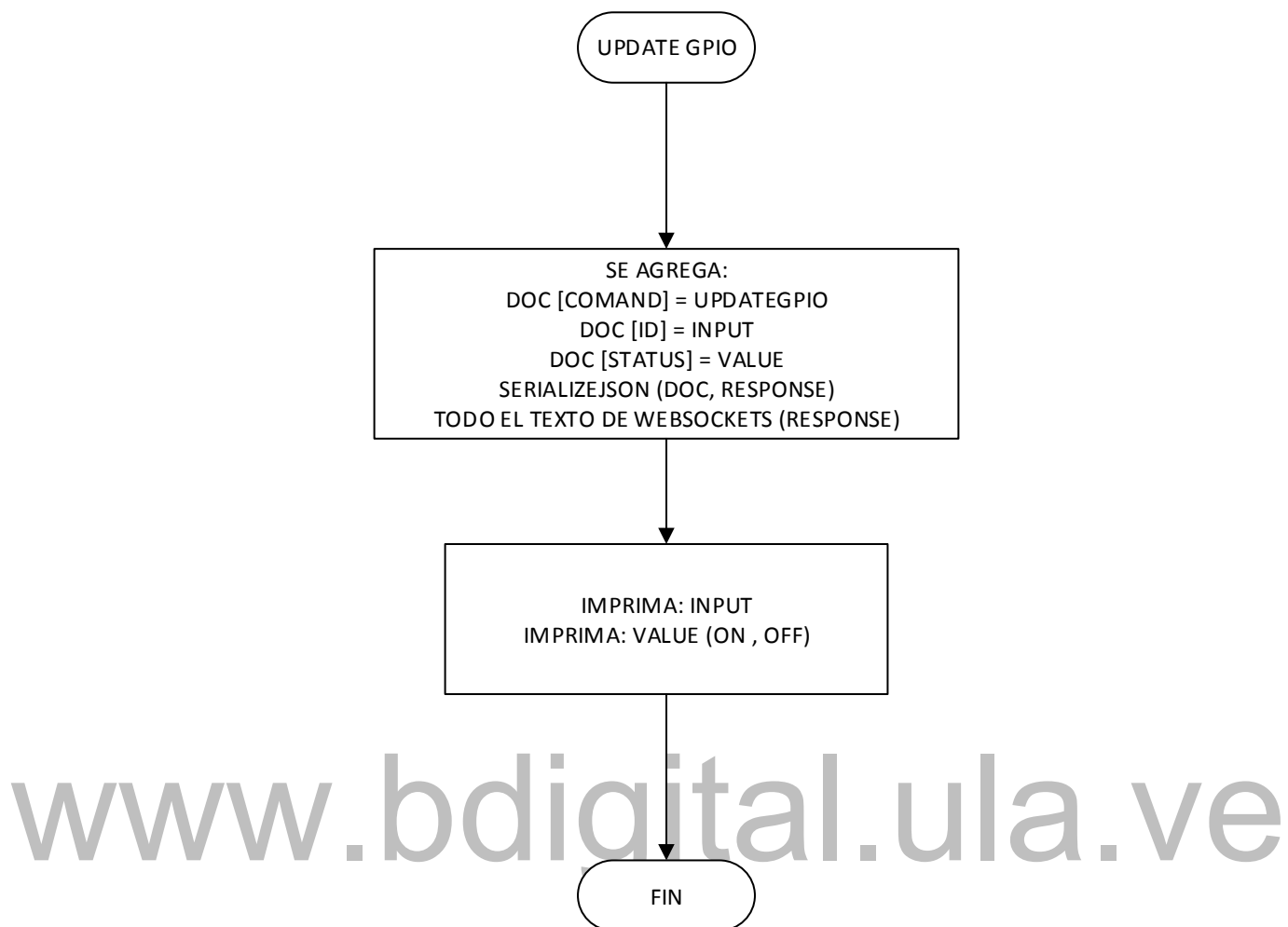


Figura 4.22 Código del botón para iniciar y detener la simulación.

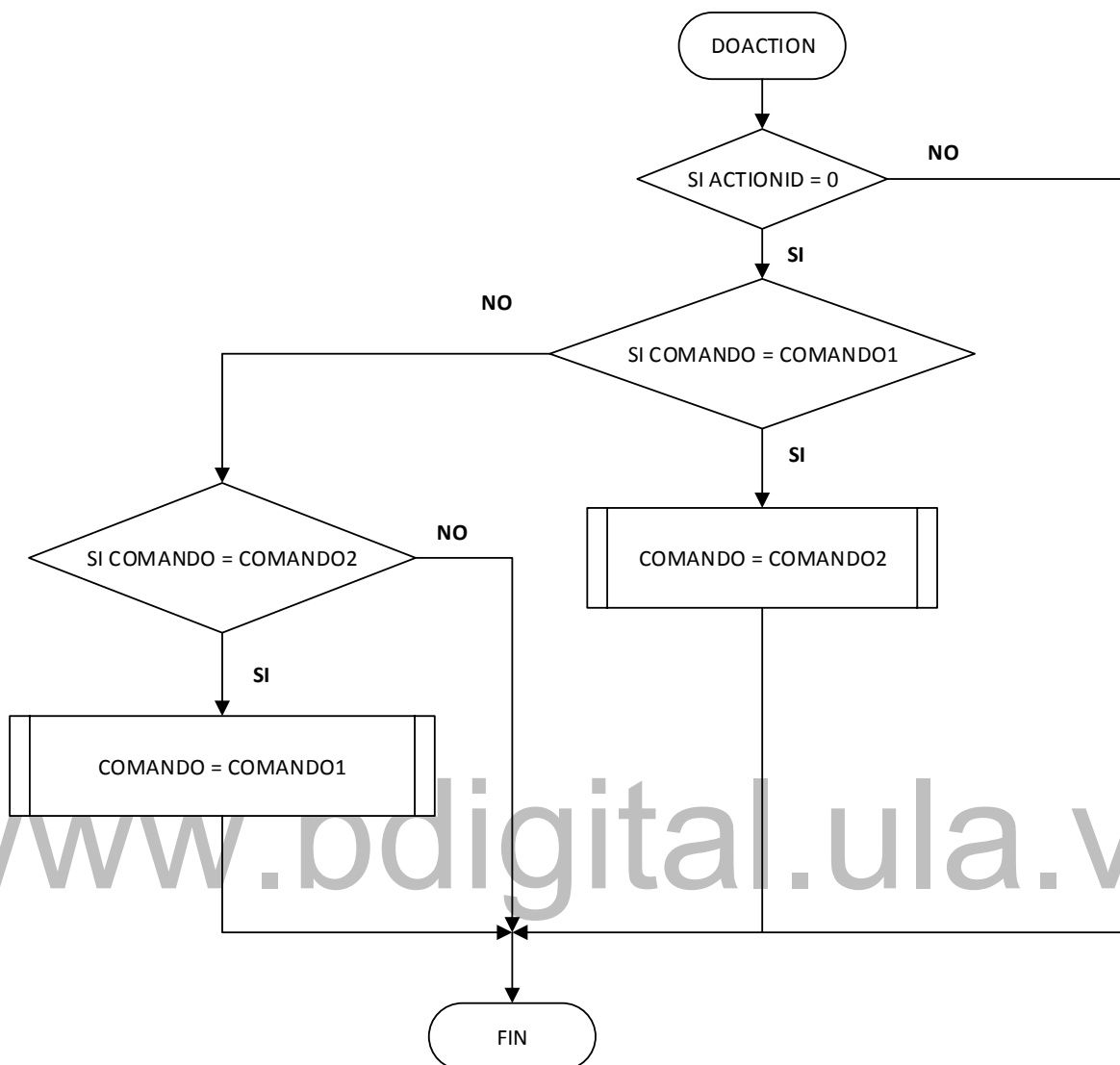


Figura 4.23 Acción de empezar y detener el programa.

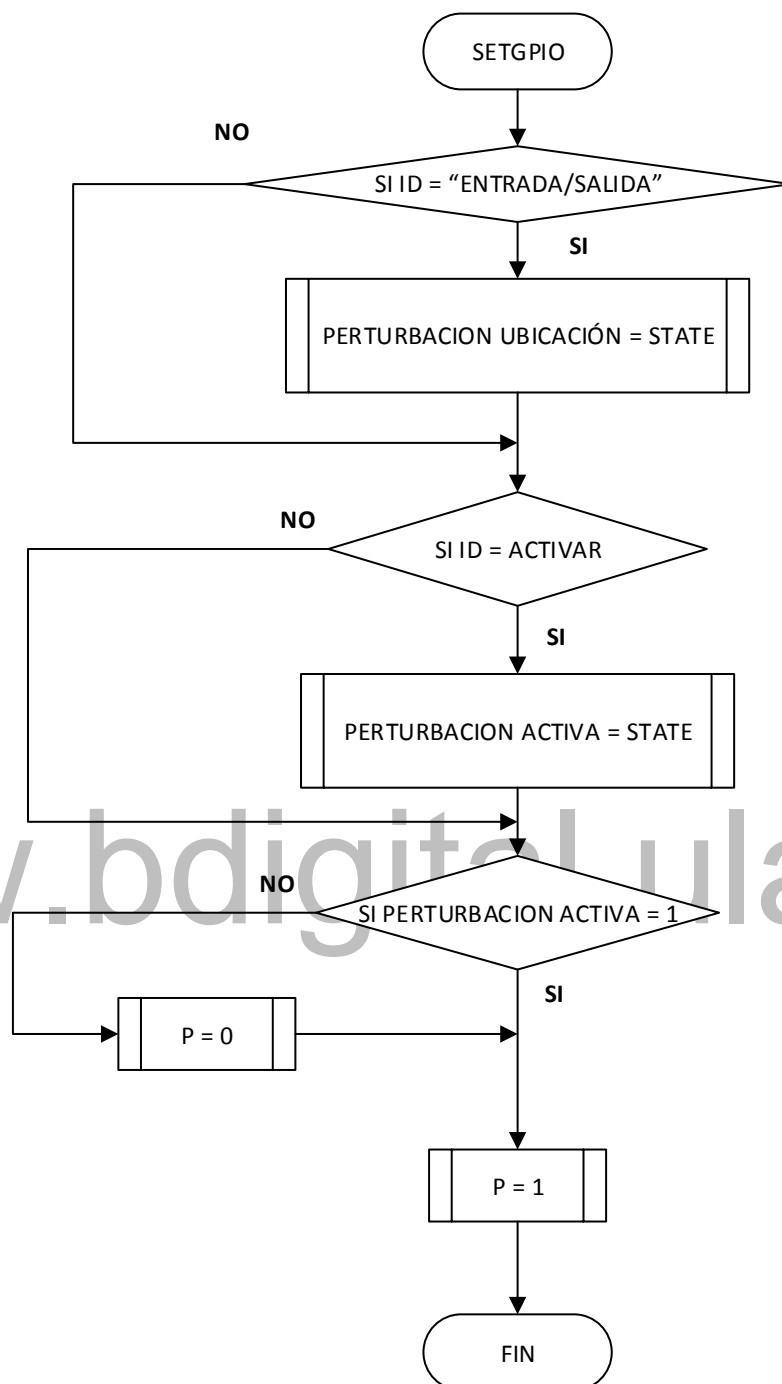


Figura 4.24 Asignación de perturbación.

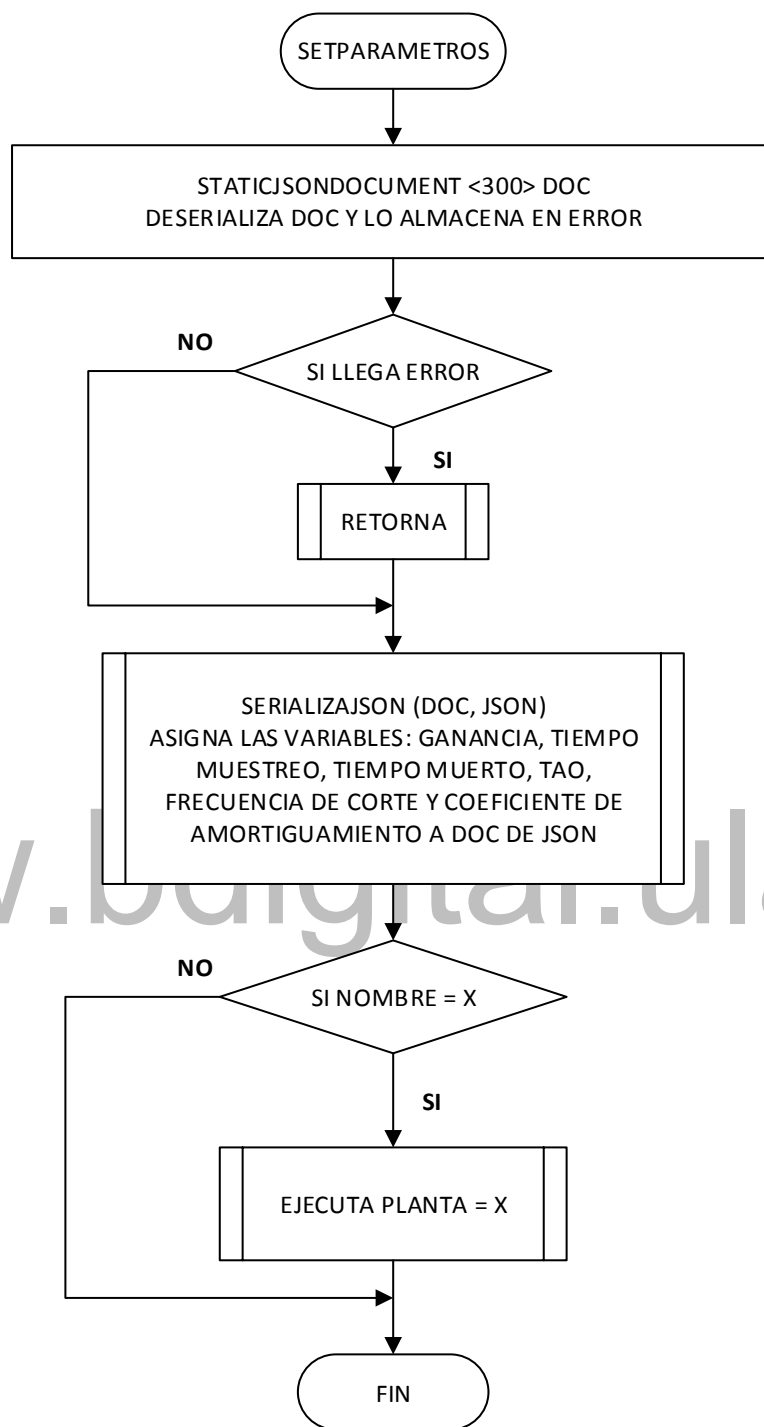


Figura 4.25 Asignación de variables de las plantas.

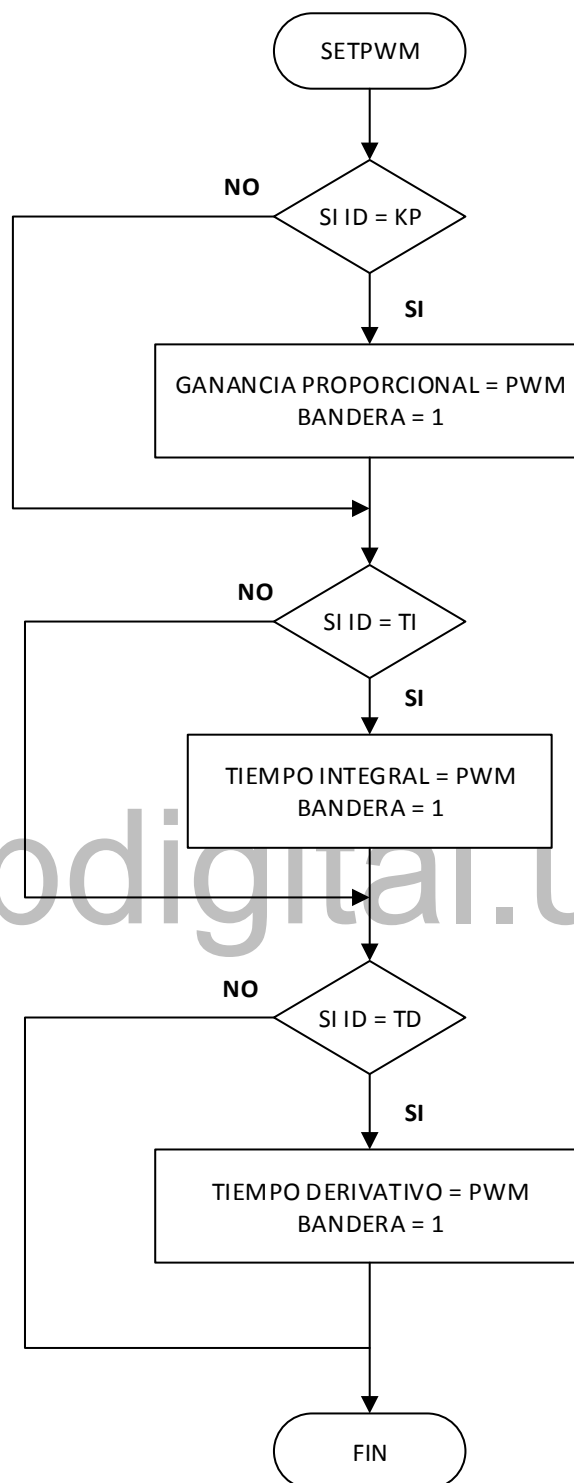


Figura 4.26 Asignación de los valores del controlador PID.

4.4 INTERFAZ GRAFICA

Con la idea de simplificar el uso de las computadoras para usuarios de todo tipo y no solo para expertos, se ha convertido en una práctica usual utilizar representaciones visuales por medio de la llamada interfaz gráfica de usuario, para que el usuario interactúe y establezca un contacto más fácil e intuitivo con el computador.

La interfaz gráfica implica la presencia de un monitor o pantalla constituida por una serie de menús e iconos que representan las opciones que el usuario puede tomar dentro del sistema, de tal manera que le sea más sencillo de utilizar y que un simple clic de ratón sobre algún gráfico, imagen, slider que aparezca en la pantalla, suplante la fastidiosa tarea de escribir un código fuente que interprete la acción que se quiera realizar.

Para este trabajo se utiliza como interfaz gráfica una página web, ya que el micro ESP32 puede implementar un servidor web para así aprovechar todo lo que el micro ofrece y la mayor razón es porque la página web pasaría a ser una interfaz gráfica universal.

4.4.1 Diseño de página web.

El diseño web implica trabajo relacionado con *layout*, así como la producción de contenido, para este caso, el diseño de la página web se creó utilizando el lenguaje de marcado de Hipertexto como HTML, el cual permite indicar la estructura del contenido y desplegar una página web. El código que se utiliza para dar estilo a la página es CSS que en español significa (hojas de estilo en cascada). También se usa JavaScript, que es un lenguaje de secuencias de comandos que permite crear contenido de actualización dinámica, animación de imágenes y características interactivas, por ejemplo, eventos que ocurren cuando se presionan los botones o datos introducidos.

La herramienta que se utilizó para el diseño de la página web como interfaz gráfica se llama Brackets, esta herramienta de edición de texto facilita el diseño en el navegador y desarrollo de aplicaciones para usuarios con herramientas visuales en el editor para que se pueda obtener la cantidad correcta de ayuda sin obstaculizar el proceso creativo.

4.5 COMUNICACION ENTRE ESP32 Y PAGINA WEB

Los protocolos de comunicación que se usaron para la comunicación entre el servidor web y el micro ESP32 son: *Json* y *WebSocket*.

4.5.1 Notación de objeto de JavaScript (*Json*)

Los *Json* son cadenas útiles que se usan cuando se quiere transmitir datos a través de una red. Debe ser convertido a un objeto originario de JavaScript cuando se requiera acceder a sus datos, esto se debe a que JavaScript posee un objeto global *Json* que tiene los métodos disponibles para convertir entre ellos. Este formato se utiliza para distribuir datos en forma de texto y permite el intercambio de información con reciprocidad entre aplicaciones de forma más sencilla, liviana y rápida posible durante el proceso.

Como antes se menciona, los *Json* distribuye sus datos en forma de texto a través de variables, para la interfaz diseñada, los *Json* están divididos en 3 etapas para el envío y recibimiento de información: botones y *sliders*, gráfica y declaración de parámetros.

- Botones y *sliders*: *id*, *command*, *status*, *pwm* y *sp*.
- Parámetros: *Nombre*, *Ganancia*, *Tiempo_Muerto*, *Tiempo_Muestreo*, *Tao*, *Frecuencia_Corte* y *Coeficiente_Amortiguamiento*.
- Gráfica: *getSensorReadings*.



Figura 4.27 Diagrama de secuencia de intercambio de mensajes.

4.5.2 Protocolo de comunicación *WebSockets*

Es un canal de comunicación abierto, en la cual queda una conexión activa tras el *handshake* inicial entre el cliente y el servidor de tal manera que el servidor pueda también enviar información nueva al cliente y recibir respuestas controladas por eventos sin que este tenga que solicitarlo previamente cada vez.

4.6 IMPLEMENTACION DEL CODIGO JAVASCRIPT

El lenguaje de secuencias de JavaScript, es un lenguaje de programación para navegadores que permite controlar multimedia, animación de imágenes y agregar características interactivas en el sitio web, como botones, sliders y datos que se introducen en formularios. Esta herramienta permite protocolos de comunicación, en este trabajo de grado se utilizarán dos: *WebSockets* y *Json*. A continuación, se mostrará la implementación del código de JavaScript por medio de diagramas de flujo.

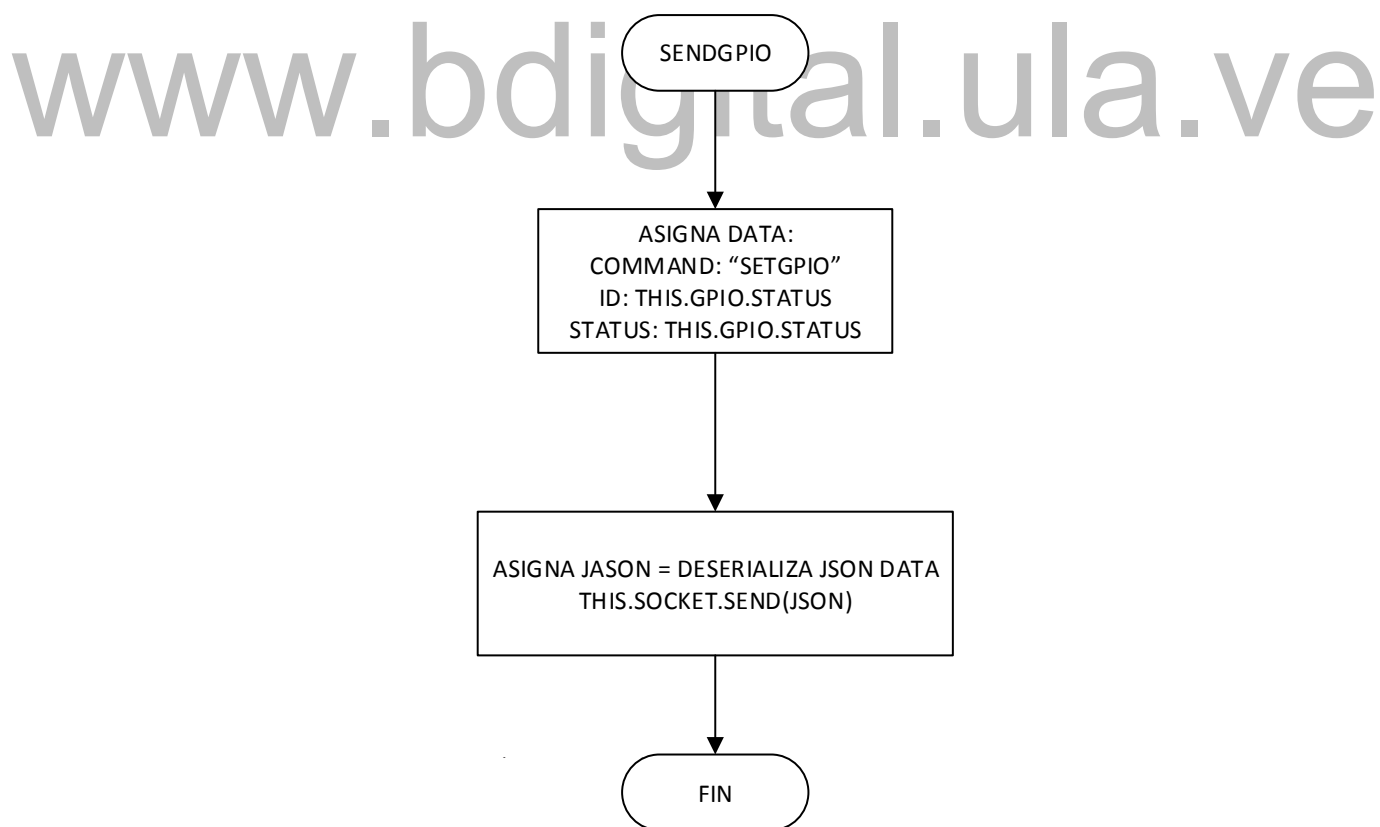


Figura 4.28 Envío del status del gpio.

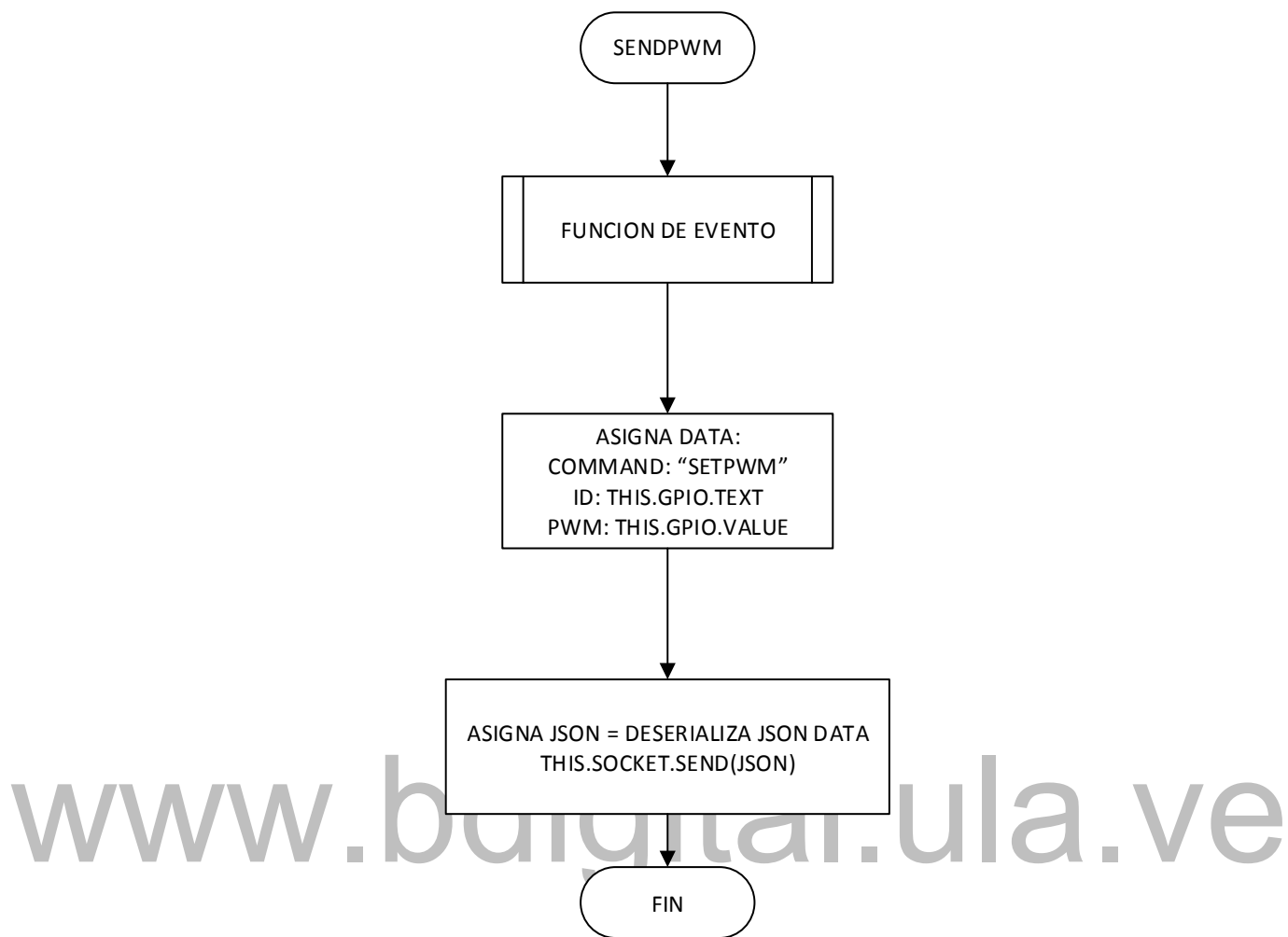


Figura 4.29 Envío de los valores del controlador PID.

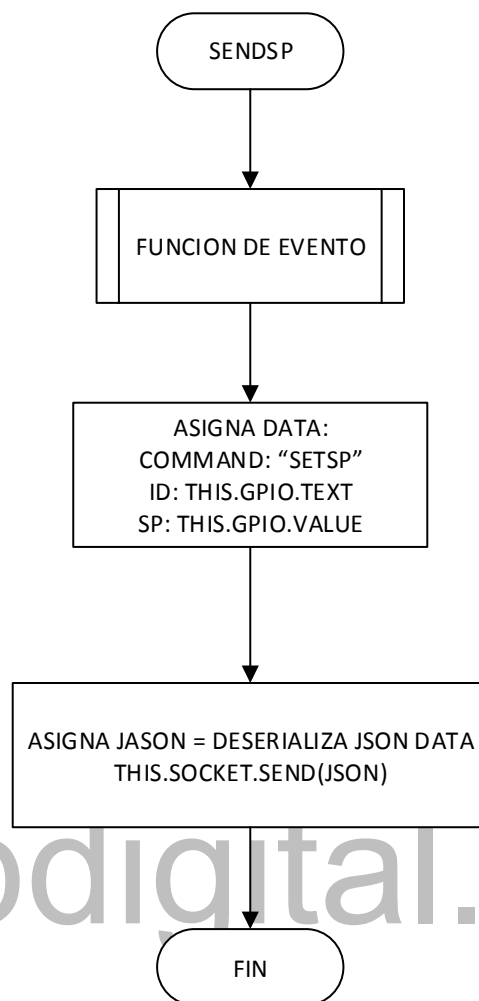


Figura 4.30 Envío del valor de la referencia a través de *sliders*.

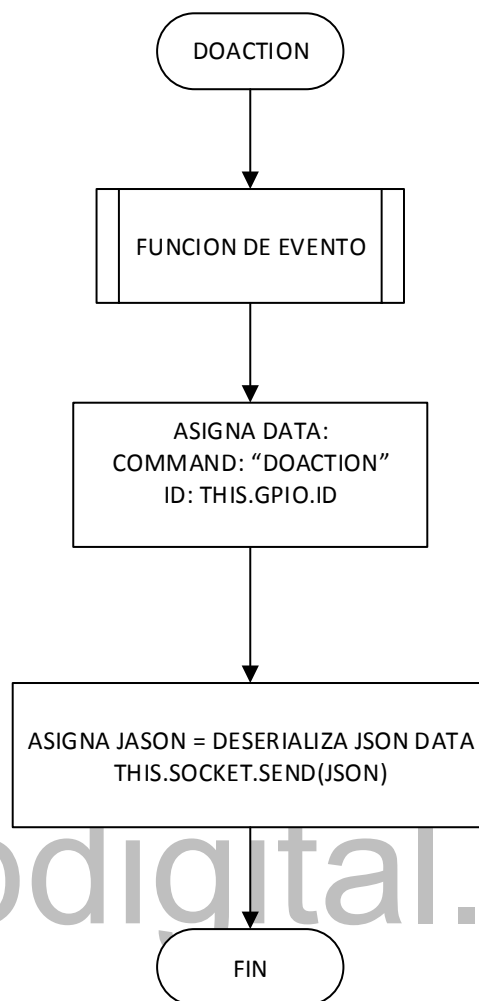


Figura 4.31 Envío del id de *doaction*.

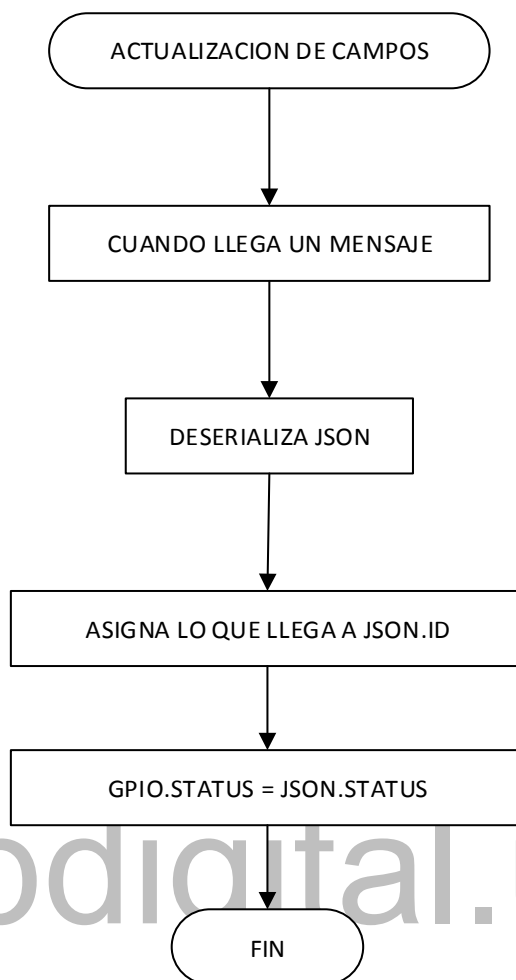


Figura 4.32 Actualización de los campos y asignación de lo que llega al *Json*.

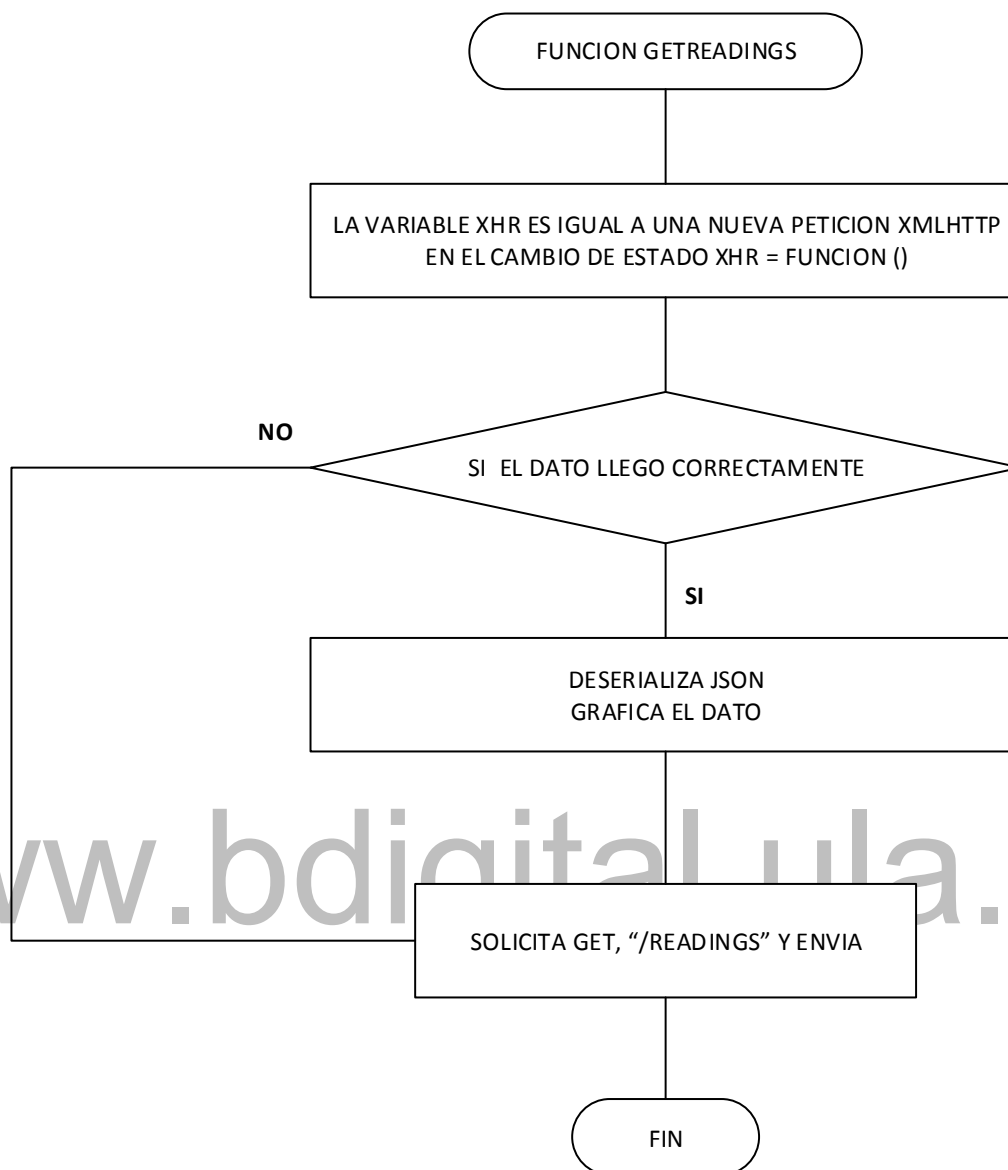


Figura 4.33 Evento de nuevas lecturas y envío por *Json*.

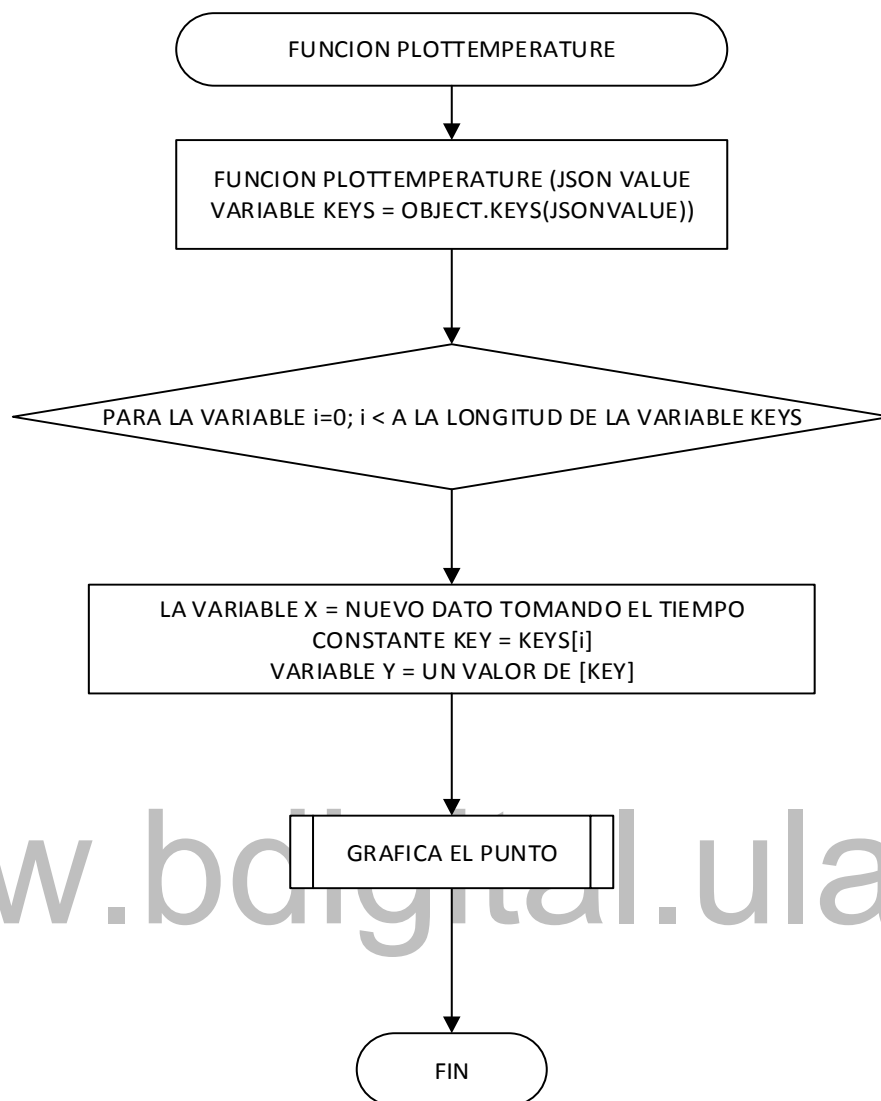


Figura 4.34 Evento del sensor de temperatura para la gráfica.

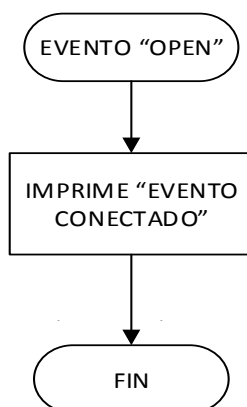


Figura 4.35 Evento *open* de conexión de la página.



Figura 4.36 Evento de erro al desconectarse la página.

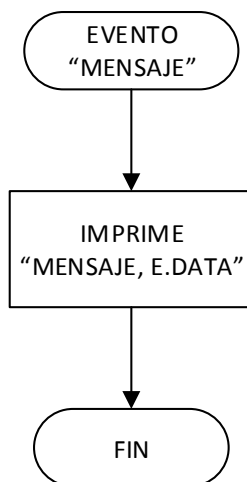


Figura 4.37 Impresión de mensajes por *JavaScript*.

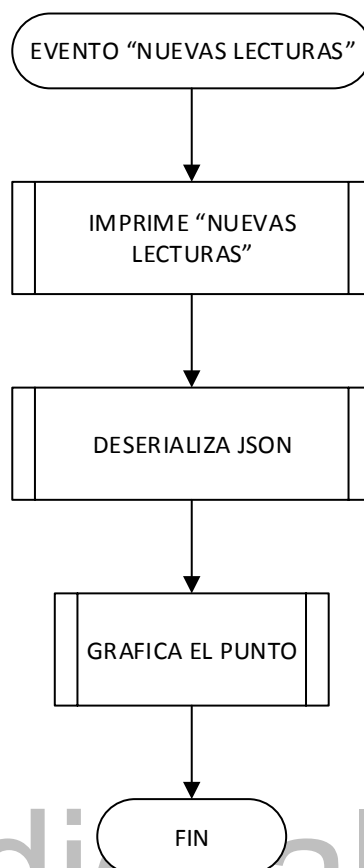


Figura 4.38 Evento de nuevas lecturas y grafica de los puntos.

www.bdigital.ula.ve

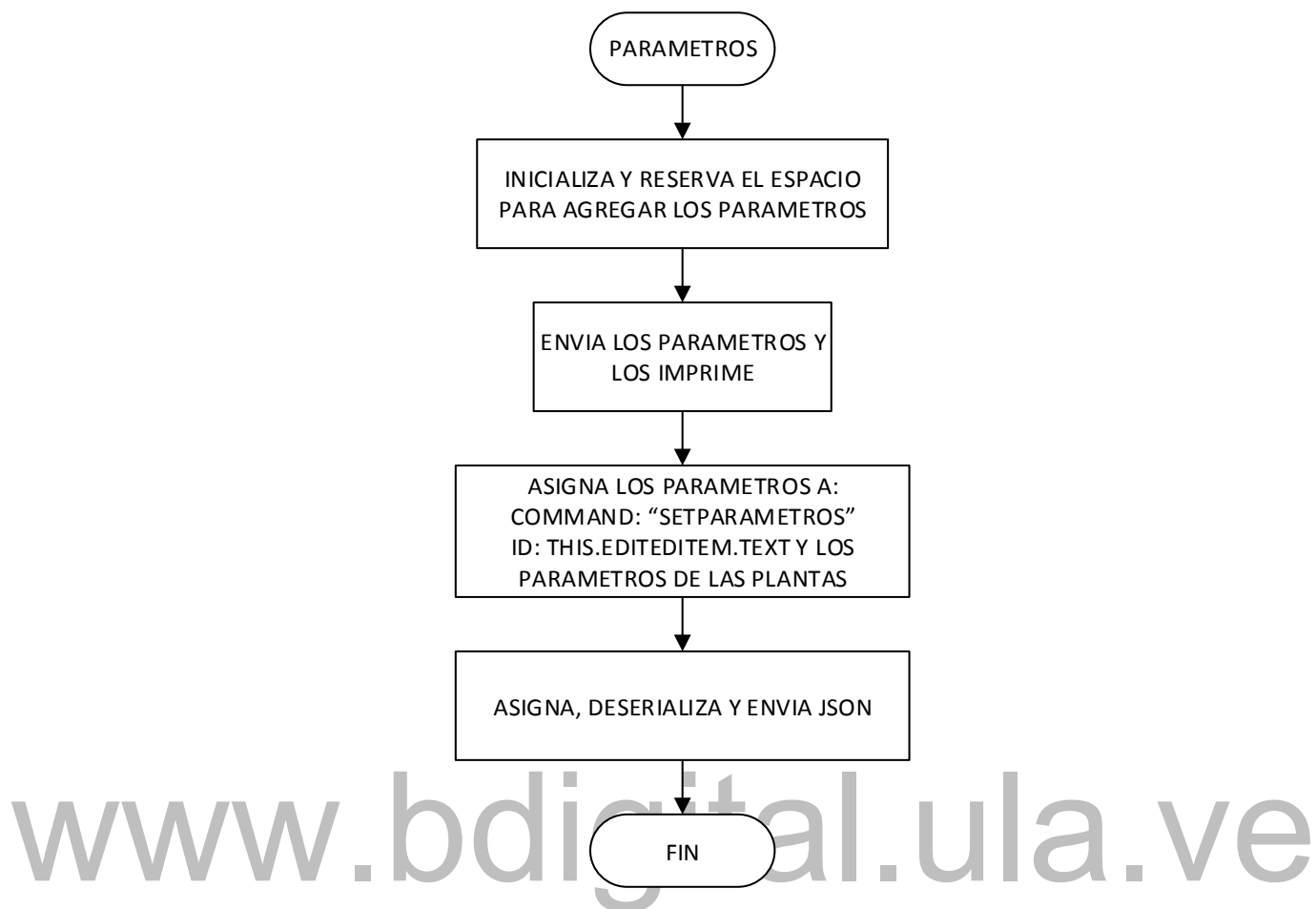


Figura 4.39 Asignación y envío de parámetros por *Json*.

CAPITULO 5

ANALISIS DE RESULTADOS

En este capítulo se ilustrará los resultados del módulo educacional usando diferentes tipos de plantas reportadas en la literatura.

5.1 RESULTADO DEL MODULO EDUCACIONAL

A continuación, se mostrará el resultado del módulo educacional, distribución del mismo para su uso, inicialización de comunicación por *WebSocket*, y simulación de funcionamiento del controlador con la referencia de salida.

```
www | 17:37:53.543 -> SPIFFS mounted successfully  
17:37:53.543 ->  
17:37:53.750 -> .....  
17:37:55.752 -> Iniciado STA: Robotica  
17:37:55.752 -> IP address: 192.168.1.120  
17:37:55.752 -> HTTP server started  
17:37:55.752 -> WebSocket server started  
17:37:55.752 -> Alarma:15 OFF  
la.ve
```

Figura 5.1 Comunicación *WebSockets* establecida.

En la Figura 5.1 muestra que la comunicación *WebSocket* está establecida, esto se observa por monitor serial, como también enseña el mensaje indicando que la pagina ha sido cargada satisfactoriamente, conexión con el servidor, dirección IP la cual se estableció para que sea siempre 192.168.1.120.

Al obtener los mensajes descritos anteriormente por monitor serial, se procede a ingresar la dirección IP en el navegador, obteniendo la página web que se muestra en la Figura 5.2, esta interfaz gráfica es de muy fácil manejo, posee cuatro botones los cual me permiten ingresar los parámetros de la planta que se quiera utilizar, registrando la planta y sus parámetros, para cargar estos valores al sistema se pulsa el botón enviar una vez tenga el registro del mismo. Los parámetros del controlador y la referencia se registran a través de sliders. Las perturbaciones se activan por medio de un *switch*, aunque el sistema no inicia hasta que se pulse el botón para empezar y este mismo sirve para detener la simulación.

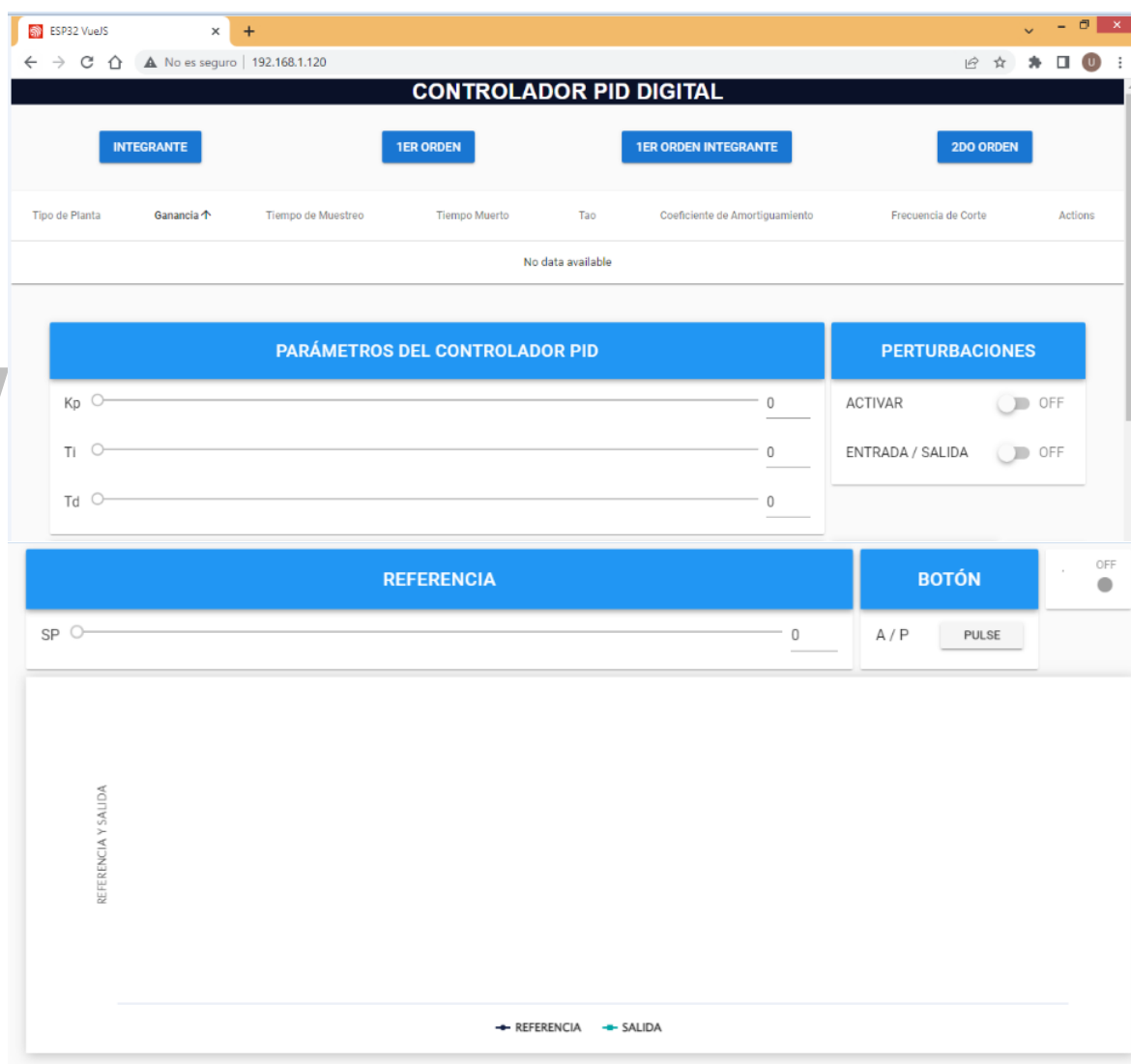


Figura 5.2 Interfaz grafica con página web.

La Figura 5.3 representa la ventana emergente que se abre al escoger la planta, esta ventana que se muestra en esta figura es para una planta integrante y tal como se puede observar tiene los parámetros de τ_o , coeficiente de amortiguamiento y frecuencia de corte bloqueados, esto se debe a que la planta integrante no posee estos parámetros, la finalidad de este bloqueo es para que los usuarios no carguen al sistema parámetros que no existen y solo puedan ingresar los valores correspondientes a cada planta.

Ingrese los Parametros

INTEGRANTE
1

Ganancia

Tiempo de Muestreo

Tiempo Muerto

Tao

Coeficiente de Amortiguami...

Frecuencia de Corte

CANCELAR GUARDAR

Figura 5.3 Ventana emergente para ingresar parámetros de la planta integrante.

Ingrese los Parametros

1er ORDEN
2

Ganancia

Tiempo de Muestreo

Tiempo Muerto

Tao

Coeficiente de Amortiguami...

Frecuencia de Corte

CANCELAR GUARDAR

Figura 5.4 Ventana emergente para ingresar los parámetros de la planta de primer orden.

Ingrese los Parametros

2do ORDEN
4

Ganancia

Tiempo de Muestreo

Tiempo Muerto

Tao

Coefficiente de Amortiguami...

Frecuencia de Corte

CANCELAR GUARDAR

Figura 5.5 Ventana emergente para ingresar parámetros de la planta de segundo orden.

Para el ingreso de los parámetros de la planta de primer orden y de primer orden integrante, la Figura 5.4 muestra la representación de la ventana emergente para estas plantas, ambas ventanas son las mismas, ya que los parámetros para estas plantas son iguales. Mientras que la planta de segundo orden no posee tao, pero sí coeficiente de amortiguamiento y frecuencia de corte, por este motivo, para esta planta, el único parámetro que está bloqueado es tao, tal y como se puede ver en la Figura 5.5.

Tipo de Planta	Ganancia ↑	Tiempo de Muestreo	Tiempo Muerto	Tao	Coefficiente de Amortiguamiento	Frecuencia de Corte	Actions
1	1	0.1	0.1				✓ ✘
4	1	0.2	0.2		2	1	✓ ✘
2	2	0.1	0.1	1			✓ ✘
3	10	0.5	1	3			✓ ✘

Figura 5.6 Registro de plantas y parámetros.

La cantidad de plantas que se pueden guardar en el registro que se muestra en la Figura 5.6 es ilimitada, pero solo se puede cargar al sistema una planta a la vez con sus respectivos parámetros, esto bajo cierto procedimiento que se mencionara más adelante.

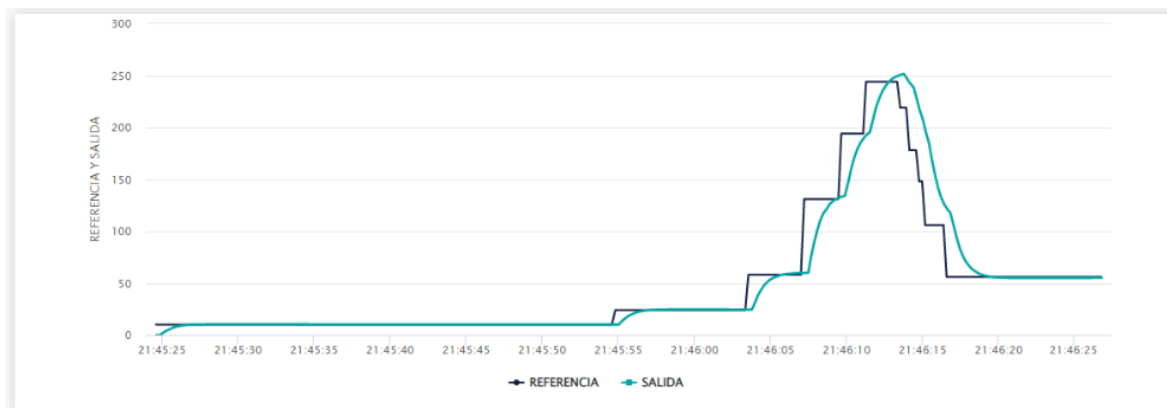


Figura 5.7 Simulación del controlador PID con la referencia.

La Figura 5.7 muestra la simulación completa del sistema, esta simulación es para una planta integrante, la línea de color azul oscuro representa la referencia y la turquesa es el controlador PID, la forma de onda que se observa del controlador al mover la referencia por medio del slider, muestra y garantiza que el controlador funciona, ya que se superpone a la referencia del sistema.

5.2 EJEMPLO 1

El sistema que se muestra a continuación se obtuvo del artículo [7] del profesor Oscar Camacho. Se considera el proceso de la función de transferencia de una planta de primer orden integrante más tiempo muerto $G(s)$ cuyos valores de los parámetros de la planta son: $K=1$, $\tau = 3$ y $t_0= 6$.

$$G(s) = \frac{1}{s(3s + 1)} e^{-6s} \quad (5.1)$$

Los parámetros de sintonización del controlador PID se obtienen a través de las ecuaciones (5.2 y (5.3 para el cálculo de la ganancia del controlador K_p , tiempo integral T_i y el tiempo derivativo T_d , que para este ejemplo es cero según el método de Minimum IAE Shinkey (1994).

$$K_p = \frac{0.556}{K(\tau + t_0)} e^{-6s} \quad (5.2)$$

$$Ti = 3.7(\tau + t_0) \quad (5.3)$$

Al sintonizar los parámetros del controlador PID por el método de Shinkey se adquieren los siguientes valores: $K_p = 0,061778$, $T_i = 33,3$ y $T_d = 0$.

El resultado del sistema para el ejemplo 1 se observa en la Figura 5.8, la cual muestra la salida del proceso y como el controlador actúa sobre la referencia de salida del sistema.

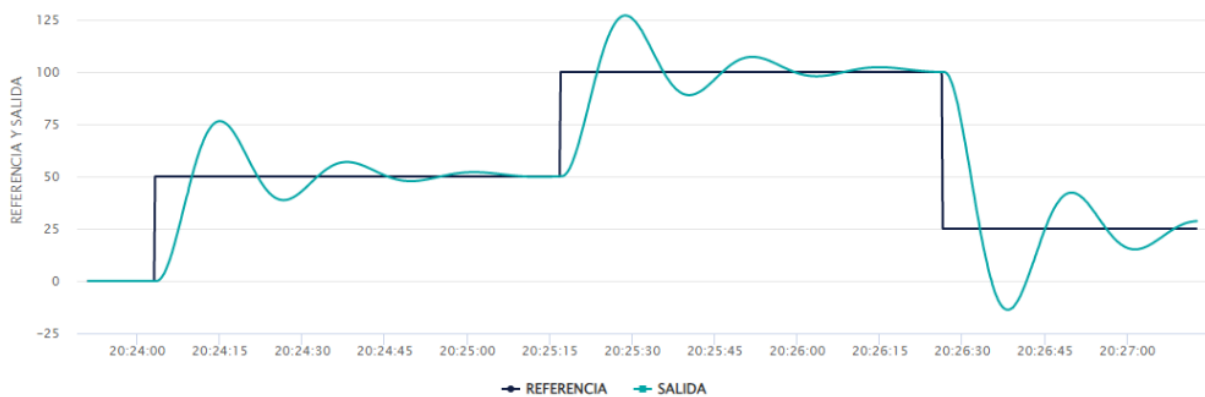


Figura 5.8 Respuesta de sistema sintonizado por el método de Shinkey.

5.3 EJEMPLO 2

El sistema para el ejemplo 2 es una planta de segundo orden, este ejemplo se puede encontrar en el artículo del profesor Oscar Camacho [8] el cual dice: considere un proceso con la función de transferencia $G(s)$ (ecuación (5.4)).

$$G(s) = \frac{-0.78}{(2s + 1)} e^{-5s} \quad (5.4)$$

Al sintonizar los parámetros del controlador PID por el método de Ziegler-Nichols se obtienen los siguientes valores: $K_p = -0.46$, $T_i = 10$ y $T_d = 2.5$.

El resultado del sistema se muestra en la Figura 5.9 y como se puede ver, la respuesta del controlador es muy lenta por lo tanto se requiere de ajustar los parámetros para que la velocidad del controlador aumente.

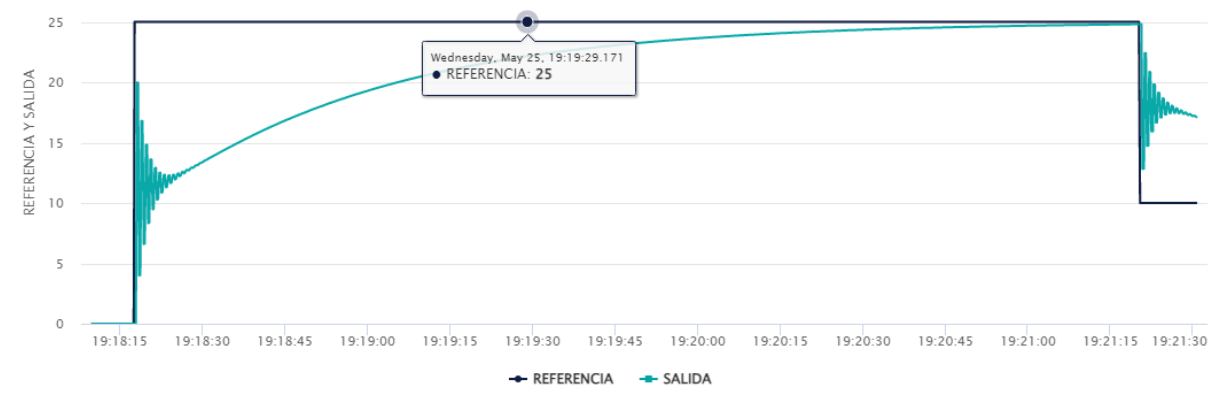


Figura 5.9 Respuesta del sistema sintonizado por el método de Ziegler-Nichols.

Para el ajuste de los parámetros del controlador, se realizaron varias pruebas para ver cuál es la más adecuada para el sistema, como se muestra en la Figura 5.10 para el primer punto de la referencia en 10, los parámetros que se usaron son los obtenidos de la sintonización, para el siguiente punto de referencia ubicado en 20, se utiliza un valor de $K_p = -1$, $T_i = 5$ y $T_d = 2$ y se observa que el controlador es más agresivo al disminuir el tiempo integral. Se modificaron nuevamente los parámetros del controlador en el tercer cambio de referencia a 35, se disminuye el tiempo integral a 1 y los otros parámetros se convierten en los mismos valores, y la respuesta aumenta siendo más efectiva, ya que aumenta la velocidad del controlador. La perturbación que se ve en la referencia ubicada en 5, se debe a que se aumentó el tiempo derivativo a 3, de manera de aumentar la agresividad del sistema, al no funcionar, se volvió a su parámetro original y se observa la misma respuesta que en el punto 3.

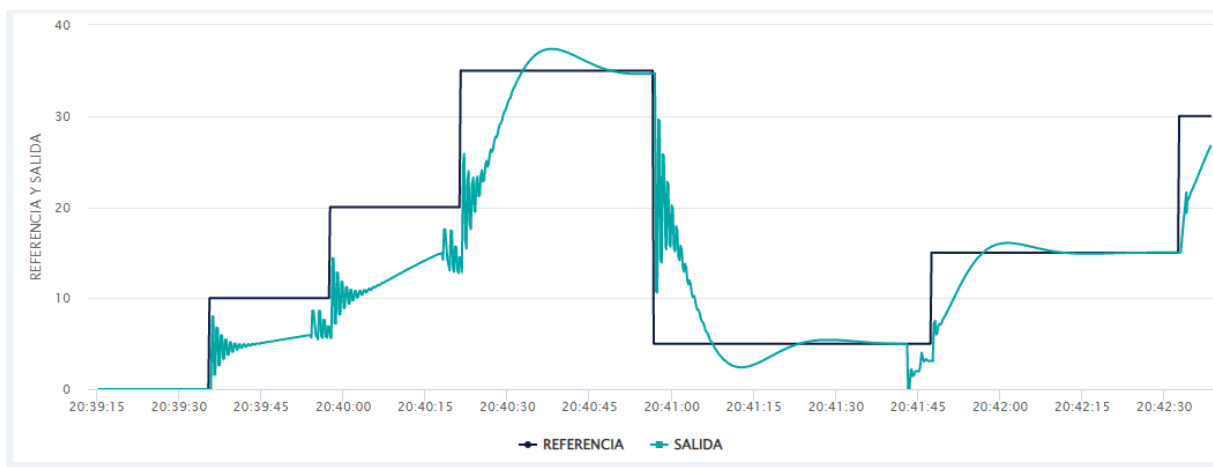


Figura 5.10 Respuesta del sistema ajustando los parámetros.

Luego de hacer varias pruebas ajustando los parámetros, se concluye que la mejor respuesta para este sistema es la que muestra la Figura 5.11, los valores de los parámetros del controlador que se usaron son: $K_p = -1$, $T_i = 1$ y $T_d = 1$. Se toma como la mejor respuesta debido a que el controlador no aumenta tanto a diferencia del de la Figura 5.10.

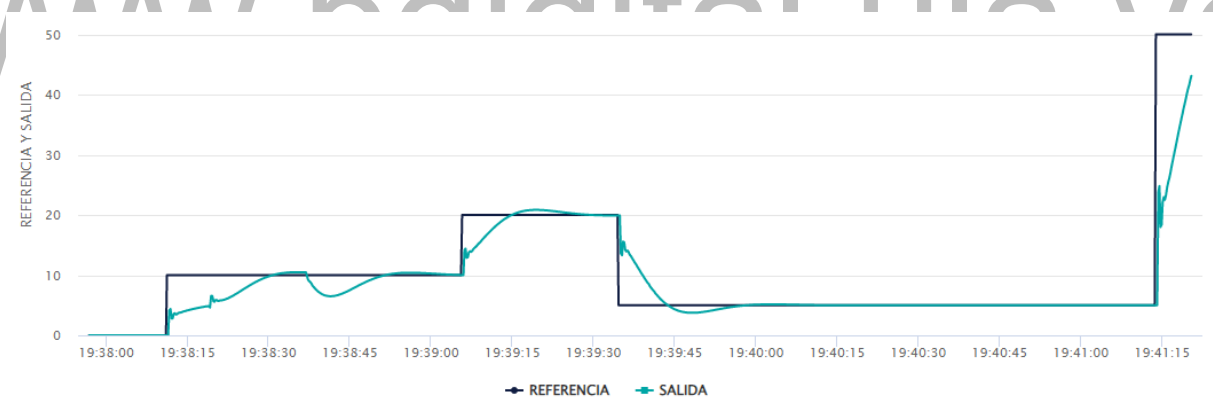


Figura 5.11 Respuesta del sistema ajustando los parámetros.

CONCLUSIONES

En este trabajo se logró cumplir y concretar los objetivos propuestos para este trabajo de grado, los cuales han sido cumplidos con éxito, al implementar el diseño de un módulo educacional para sistemas de control funcional, que emula los procesos de 4 plantas y un controlador PID de forma digital, que, al ser comparados con los sistemas reales, las simulaciones de este módulo coinciden con el sistema real.

Se realizó el estudio de la teoría de controladores PID, sintonización de los mismos y discretización de los sistemas. Para la discretización se observa que afecta la respuesta y el desempeño en comparación al análisis en tiempo continuo, principalmente en el tiempo muerto, por este motivo se desechó el método de aproximaciones y se sustituyó por desplazamientos con líneas de retardo obteniendo así la respuesta deseada con los sistemas discretizados.

Para el desarrollo de este emulador, las simulaciones a través de la herramienta Matlab Simulink fueron clave durante el proceso, ya que fueron el punto de referencia para comprobar la veracidad y el funcionamiento del programa en el micro.

El diseño de este controlador se observa por medio de una interfaz (página web) de fácil uso, la cual funciona correctamente, aunque la simulación del proceso depende de la tarjeta de video de la computadora en la que se esté ejecutando, ya que, si la velocidad de la tarjeta es baja, la simulación tiende a pegarse, pero no afecta el funcionamiento interno ni la veracidad del programa.

RECOMENDACIONES

Con base a los resultados obtenidos en este proyecto, se desea mejora del mismo, por lo tanto, se realizan las siguientes recomendaciones:

- Aunque las 4 plantas utilizadas en este proyecto son las más utilizadas a nivel industrial, se recomienda agregar más tipos de plantas, de tal manera, que este emulador sea más completo en cuanto a tipos de modelados.
- Permitir la simulación de varias plantas al mismo tiempo, con los diferentes parámetros de control y planta que presenten cada una de ellas.
- Agregar más perturbaciones al sistema y ubicarlos en diferentes puntos a través del mismo.
- Permitir al servidor que trabajen en la simulación varios usuarios con sistemas independientes.
- Permitir el acceso a varios usuarios a la vez, donde todos puedan ver la misma simulación en tiempo real, de tal manera que se pueda dictar una clase a través del emulador simulando un proceso y que todos los alumnos puedan conectarse a través de sus dispositivos y observar al mismo tiempo la respuesta del controlador.
- Agregar un sistema de alarma con sonido para que al momento de que ocurra un error o desborde de información en el micro, el usuario presencie esta falla.
- Integrar el “*anti –reset wind up*” al código del emulador para restringir el incremento del término integral en ciertas plantas, como, por ejemplo: un controlador neumático, el cual este efecto se evidencia en el fuelle integral, que se infla hasta su máximo.

REFERENCIAS

- [1] «Retardo o Tiempo Muerto de un Sistema,» 14 Septiembre 2021. [En línea]. Available: <https://controlautomaticoeducacion.com/analisis-de-sistemas/retardo-o-tiempo-muerto-de-un-sistema/>. [Último acceso: 2021 Octubre 20].
- [2] A. H. Picerno, «El Algoritmo PID,» *SABER ELECTRONICA*, vol. 175, n° 60, p. 79, 2005.
- [3] F. Ventura Gutierrez, «Anti Reset - wind up,» *Clud Saber Electronica*, vol. 18, n° 19, p. 75, 2006.
- [4] L. Moreno, G. Santiago y C. Balaguer, *Ingenieria de control*, Barcelona España: Ariel, S.A., 2003.
- [5] O. Katsihiko, *Ingenieria de control moderna*, Madrid España: PEARSON EDUCACION, S.A., 2010.
- [6] S. Toledo, *Diseño de controladores PID en tiempo discreto, y analisis de respuesta utilizando herramientas computacionales*, Guatemala, 2007.
- [7] Camacho, Oscar ; De la Cruz Francisco, «Smith predictor based-sliding mode controller for integranting processes with elevated dead time,» *ISA TRANSACTION* , n° 43, pp. 257-270, 2004.
- [8] Camacho, Oscar , «A PREDICTIVE APPROACH BASED-SLIDIN MODE CONTROL,» *15th Trieninial Congress, Barcelona, Spain*, 2002.

- [9] S. Cobo y J. David, «Repositorio Institucional de la Unive de las Fuerzas Armadas ESPE,» 08 Julio 2011. [En línea]. Available: <http://repositorio.espe.edu.ec/handle/21000/219>. [Último acceso: 28 Abril 2022].

www.bdigital.ula.ve

ANEXOS

www.bdigital.ula.ve

MANUAL DEL USUARIO

1. Encender el micro ESP32.
2. Comprobar por puerto serial que se realizó conexión con SPIFFS.
3. Ingresar al navegador.
4. Ingresar la dirección IP (168.192.1.120).
 - 4.1. Esperar que la pagina cargue.
5. Seleccionar el tipo de planta.
6. Ingresar los parámetros de la planta que selecciono.
 - 6.1. Presionar guardar, para guardar los parámetros y el tipo de planta que selecciono.
 - 6.2. Se puede guardar varias plantas con sus parámetros.
7. Pulsa el botón para cargar los parámetros al micro.
8. Utiliza las sliders para seleccionar los valores de los parámetros del controlador.
9. Pulse el botón para arrancar la simulación.
 - 9.1. Pulsar el botón de perturbación para activarla, se puede escoger si quieres la perturbación en la entrada o la salida con el botón de entrada/salida.
 - 9.2. Pulsar nuevamente si quieres detener la simulación.
10. Para cargar otra planta con sus parámetros, pulsar el botón de cargar los parámetros de la nueva planta con la que se quiere trabajar mientras la simulación está detenida y continua con el paso 8 y 9
11. Si se dispara la alarma, reinicia la página y espera 45 segundos para que esté nuevamente activa.